

317.057
tanulmányok

1974/18

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADEMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET

TANULMÁNYOK

18/1974.

746579 MTA KESZ Sokszorosító. F. v.: Szabó Gyula

MAGYAR
TUDOMÁNYOS AKADÉMIA
KÖNYVTÁRA

Pikler Gyula

MINI-SZÁMITÓGÉPES INTERAKTIV ALKATRÉSZPROGRAMIRÓ RENDSZER
NC SZERSZÁMGÉPEK AUTOMATIKUS PROGRAMOZÁSÁHOZ

Kandidátusi értekezés

Budapest, 1974.

TARTALOMJEGYZÉK

	Oldal
1. BEVEZETÉS	9
1.1 A dolgozat célja.	9
1.2 A koncepció jellemzése.	10
2. A TÉMA RENDSZERTECHNIKAI HELYE A SZÁMITÓGÉP SOFTWARE TUDOMÁNY KERETÉBEN	13
3. ELŐZMÉNYEK.	17
4. AZ ALKALMAZOTT TERMINOLÓGIA	23
5. AZ APT PROGRAMCSALÁD TAGJAINAK KÖZÖS TULAJDONSÁ- GAI ÉS AZ EGYES RENDSZEREK ÖSSZEHASONLÍTÁSA AZ IRODALOM ALAPJÁN.	27
6. AZ INTERAKTIVITÁS SZEREPE	33
7. SPECIÁLIS HAZAI SZEMPONTOK.	43
8. A VONATKOZÓ IRODALOM KRITIKÁJA.	45
8.1 Az APT rendszerek és nyelvek kritikája. . .	45
8.2 Az interaktív rendszerek kritikája.	52
9. INTERAKTIV RENDSZEREK TERVEZÉSÉNEK ÁLTALÁNOS SZEMPONTJAI	57

	Oldal
10. A DOLGOZATBAN JAVASOLT UJ INTERAKTIV ALKAT- RÉSZPROGRAMIRÓ RENDSZER	65
10.1 A megalapozás szempontjai.	65
10.2 A rendszer felépítése, szervezésének strukturája.	75
10.3 Az APT típusu nyelvek utasításainak fel- osztása a gráfon belül	77
10.4 A dialógusok megtervezésének szempontjai	85
11. AZ AIR ALKATRÉSZPROGRAMIRÓ RENDSZER	89
11.1 A DISTAR-B általános dialógus rendszer .	92
11.2 AIR közbelső dialógus nyelv.	99
11.3 A TRANSAIR az állapotrekordokat leképező rendszer	111
11.31 A TRANSAIR nyelv kialakításának szem- pontjai.	112
11.32 Az alkatrészprogramiró rendszerben használt dialógusok fajtái és azok felosztása	113
11.33 A TRANSAIR nyelv leírása	116
11.34 A TRANSAIR nyelven írt program szabá- lyai	121
11.35 A TRANSAIR fordító program	125
11.4 AIR dialógusprogram.	128
11.41 Az AIR programban felhasznált program- szegmensek	130
11.42 Az AIR program működése.	140
12. ÖSSZEFOGLALÁS	143

	Oldal
1. Függelék	
Az EXAPT1 nyelv utasításkészletének felosztása	145
2. Függelék	
Néhány EXAPT1 utasítás meghatározásához tervezett display dialógusképek	171
3. Függelék	
A TRANSAIR, valamint az AIR közbenső nyelveken írt dialógusprogramok.	187
4. Függelék	
Az AIR program FORTRAN listája	215
5. Függelék	
Az interaktiv alkatrészprogram írásának folyamata. A szimulációs AIR program outputja	255
IRODALOMJEGYZÉK	289

1. BEVEZETÉS

1.1 A dolgozat célja

A dolgozat egy összefoglalt kutatómunkáról, olyan kis-számítógépes interaktív rendszer létrehozásáról számol be, amely a számjegy vezérlésű szerszámgépek APT típusu alkatrészprogramozásának új, automatizált megközelítését valósítja meg. A dolgozatban bemutatjuk az eddig hasonló céllal, de más koncepció alapján készült rendszereket, majd az új rendszert az előzőekkel összehasonlítva ismertetjük, hogyan sikerült az új eredményeket elérni.

A most megvalósított rendszer, az AIR (APT típusu Interaktív alkatrészprogramíró Rendszer) jellemzője az, hogy lehetővé teszi az APT programcsalád bármely tagjára interaktív módon alkatrészprogramok írását. Megalkotásának módja olyan, hogy a belső struktúra a feladat bonyolultsága ellenére is relative egyszerű. Ez lehetővé teszi azt, hogy a rendszer a mini-számítógépen valósuljon meg. Interaktív hardware-ként alfanumerikus display-t alkalmaz.

A mini-számítógépes megoldás következménye az, hogy megnöveli az ipari alkalmazás lehetőségeit. A rendszer dialógus formájú. Működése részben szintetikus jellegű, mert előre definiálatlan programokat épít fel definiált építőkövekből, részben analitikus. Az utóbbit a beépített diagnosztikai programrészek képviselik. Ezenkívül még tanító jellegű is, ami alapvető céljából következik. Az ipari alkalmazás szemszögéből lehetővé teszi azt, hogy kevésbé gyakorlott programozók jó NC alkatrészprogramot írjanak. Használata következtében csökken a szubjektív programozási hibák valószínűsége.

1.2 A koncepció jellemzése

Az ismertetett koncepció részben az eddigiektől eltérő, új programozási struktúra megalkotását tette szükségessé, figyelembe véve az APT típusu alkatrészprogramírásának munkapszichológiai elemzését és annak alkalmazását.

Az AIR alkalmazásának, illetve egy tetszés szerinti, szabadon választható APT típusu nyelvre adaptálásának előre látható nehézségeit úgy küszöböltük ki, hogy megírtuk a dolgozatban ugyancsak ismertetett TRANSAIR fordítóprogramot. A TRANSAIR inputja egy dialógus nyelven megírt program, outputja pedig állapotrekordok halmaza, amelyet a DISTAR-B általános dialógus rendszer dolgoz fel. A DISTAR-B rendszer az állapotrekordokat táblázatba állítja össze, amelynek strukturája alkalmas arra, hogy az AIR program azt felhasználja. Minden egyes APT típusu nyelv konkrét feladataihoz, például az EXAPT1-hez, a 2CL-hez stb., más és más állapotrekord tábla tartozik. A TRANSAIR és a DISTAR-B rendszerek működés szempontjából függetlenek az AIR rendszertől. Feladatuk az állapottáblák összeállítása, amelyek alapján az AIR program az alkatrészprogram írásához szükséges dialógusokat végrehajtja. Így válik az AIR rendszer az APT alapu nyelvek körében általánossá.

A példaként kivitelezett rendszer két változatban készült el - 1010B kisseámítógépen és CDC típusu közepes számítógépen. Az utóbbi a kisseámítógépes rendszer szimulációja FORTRAN-IV-ben, az algoritmus ellenőrzésére és leírására.

A koncepció jellemzésére végül meg kell említeni, hogy az AIR rendszer kishámitógépen valósult meg, ami alapvető programszervezési eljárást igényelt. Ez messze tulmegy azon az egyszerű módszeren, hogy a programot szegmentáljuk és a részprogramokat overlay szervezés- sel kapcsoljuk össze. Erre a problémakörre itt a beve- zetőben csak utalunk. A kidolgozott eredményeket és megoldási módokat a következő fejezetekben ismertet- jük.

2. A TÉMA RENDSZERTECHNIKAI HELYE A SZÁMITÓGÉP SOFTWARE TUDOMÁNY KERETÉBEN

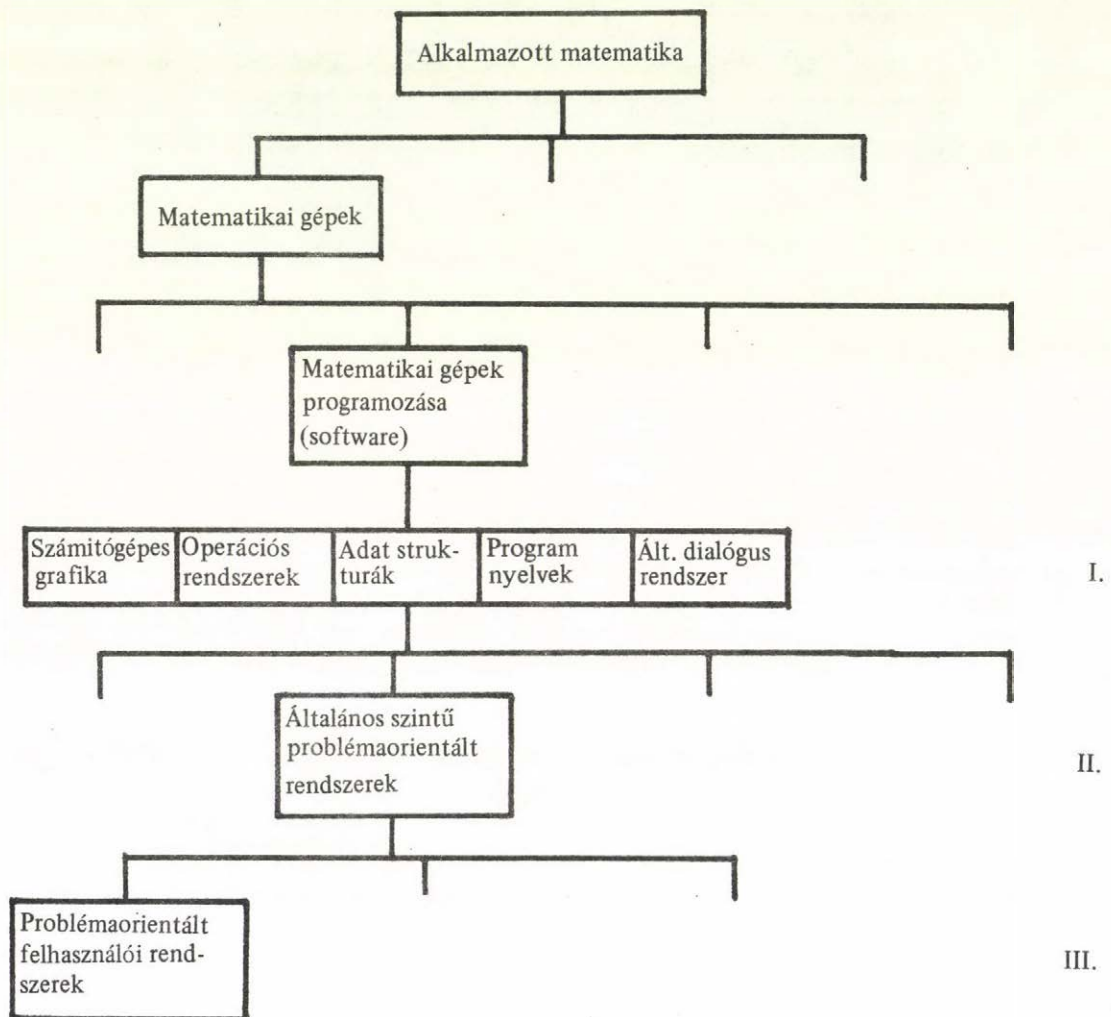
A számítógép tudomány általános rendszerezésében vannak bizonyos elfogadott konvenciók. Ezek szerint különválasztják a hardware és a software fejlesztésében szükséges tudományos alapok művelését. Tekintve, hogy a dolgozat kizárólag a software terén nyújt újat, a hardware-t pedig csak eszközként használja fel, rendszertехnikai besorolásban a software fejezetbe tartozik, amely az alkalmazott matematikai tudomány egyik ága.

Az ismert felosztások közül a közelebbi besorolásra azt a többszintes analitikus rendszert vesszük alapul, amely az általános software-ből indul ki és felosztásban a problémaorientált felé halad.

A software kutatás egyes szintjeit, illetve a kialakított rendszerek egymáshoz való kapcsolódását az 1. ábra egyszerűsített vázlata mutatja.

Az 1. ábra I-gyel jelölt szintjén helyezkednek el az általános programozási rendszerek, másképp kifejezve a programozási rendszereket felépítő magas szintű rendszerek. Ilyenek például az operációs rendszerek, az általános szintű adatstrukturák, általános szintű programozó nyelvek, dialógus rendszerek, számítógépes grafikák, makró rendszerek stb.

Az ábra következő, II. szintjén helyezkednek el az általános szintű, de egyuttal problémaorientált rendszerek. Általában ezek egy jól körülhatárolt, de széles körű al-



1. ábra

kalmazási terület problémáit fogalmazzák meg általános szinten.

A körülhatárolt alkalmazási területen belül a II. szint egy-egy részhalmozát képviseli a III. szint egy-egy kockája, tényleges problémaorientált feladatok megoldására.

A dolgozatban javasolt AIR rendszer az ábra II. szintjén helyezkedik el, mert az AIR rendszer jellemzője az, hogy bármely APT típusu nyelvre interaktív alkatrészprogramíró rendszert valósít meg és ilyen módon az általános szintű problémaorientált rendszerek családjába tartozik. Ehhez kapcsolódik a TRANSAIR és a DISTAR-B, amelyek ugyancsak a II. csoportba tartoznak, mert rendszerteknikailag problémaorientált programokat kezelő általános rendszerek.

Ebben a rendszerteknikai felosztásban a dolgozat témakörének csak fő irányvonalát vettük figyelembe. Felhasználja azonban más tudományterületek, például a munkapszichológia eredményeit, alkalmazásának ipari aspektusa szempontjából pedig a gépészeti tudományokhoz is kapcsolódik.

3. ELŐZMÉNYEK

A történeti előzményeket először az alkalmazás oldaláról közelítjük meg.

Az NC szerszámgépek első típusát az amerikai repülőgépiparban dolgozták ki, háromdimenziós pályavezérlésre, 1948-ban. [62] Az első kísérlet sikere után csak 1955-ben kezdődött az NC gépek polgári ipari alkalmazása, mert addig zárt témaként kezelték és főleg a repülőgépiparban alkalmazták. A polgári iparban a kezdeti nehézségek után az NC gépek különleges eszközökből ipari terméké váltak. Rohamos elterjedésükről számos statisztika és irodalmi adat áll rendelkezésre [64, 147, 148, 149, 139, 114, 8, 166, 88, 167, 44].

A lényeges az, hogy a fejlett ipari országok szerszámgépipara az előkészületi idő után körülbelül két évtized leforgása alatt exponenciálisan áttér az NC vezérlésre a hagyományos szerszámgépek rovására. Ez a folyamat az USA-ból elindulva áttért az európai szerszámgépiparra is. Jelentős késéssel, de intenzíven megindult az NC fejlesztés és alkalmazás a hazai iparban is. Ipari alkalmazás szempontjából ez a folyamat teszi szükségessé, hogy a fejlett ipari országokkal szembeni lemaradást pótoljuk. Ebbe a folyamatba szervesen illeszkedik bele az NC programozás fejlesztése olyan programozási eljárással, amely képes alkalmazkodni a különböző típusú NC gépek programrendszeréhez.

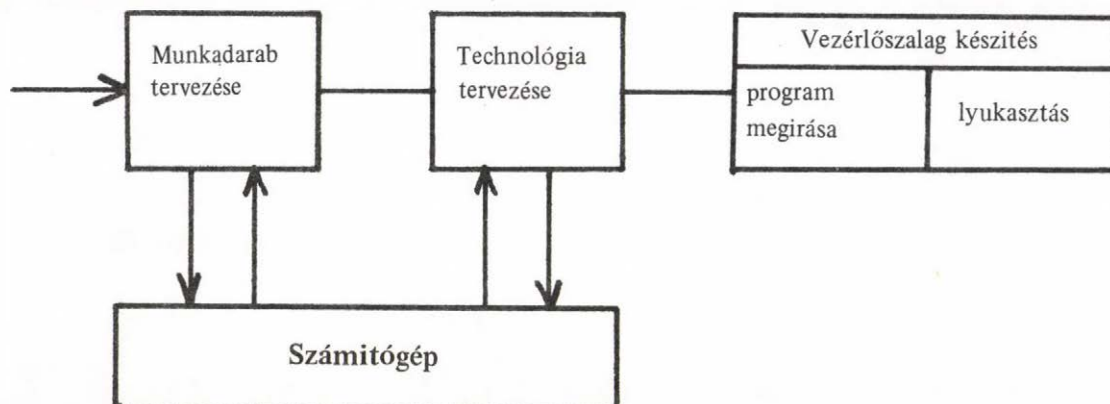
Az NC gépek fejlődéstörténetével különböző tanulmányok részletesen foglalkoznak. [12, 87, 97, 109, 127, 135, 91,

136, 98, 6, 23, 34, 164, 9, 58, 161, 84, 106, 57, 93, 18, 4, 14, 59, 61, 5, 105, 7, 164].

Hazánkban az OMFB keretében dolgozták ki azt a fejlesztési koncepciót, amely a következő évek fejlesztési irányzatát előírta. [163, 57]. Itt csak azt tartjuk lényegesnek megemlíteni, hogy az NC gépek fejlődésével fokozatosan magasabb színvonalu számítástechnikai eszközöket alkalmaznak. Ez az irányzat érvényesül a szerszámgépek vezérlésében és programozásában.

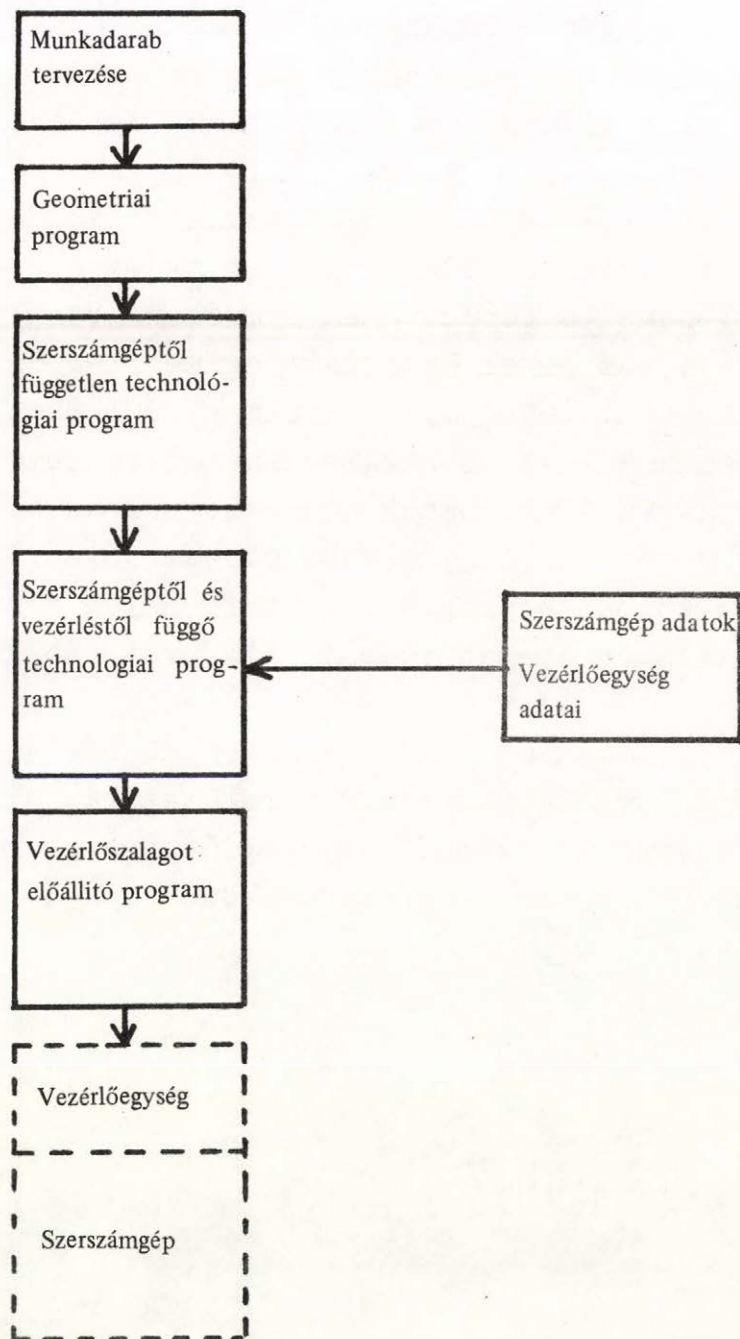
Az utóbbira térünk ki részletesebben, amely a dolgozat tárgyához közelebb áll.

Az NC szerszámgépek automatikus programozását a már hagyományossá vált manuális programozással szemben az jellemzi, hogy a számítógép veszi át a vezető szerepet. [91, 28, 33]. Összehasonlítás kedvéért megemlítjük a számítógép szerepét a manuális programozásban, amelyet az alábbi blokkdiagram mutat (2. ábra)



2. ábra

A fejlődés irányzata azonban az, hogy a technológiai tervezést, az NC vezérlő lyukszalag készítést számítógépes rendszer oldja meg. Egyszerűsített elvi séma mutatja a fejlődésnek ezt az irányzatát (3. ábra)



3. ábra

A magas szintű programozás folyamatának elindítója a MIT (Massachusetts Institute of Technology) APT (Automatically Programmed Tool) rendszere, amelynek első eredményeit 1961-ben hozták nyilvánosságra. [32] Az eredeti APT III rendszert háromdimenziós megmunkálásra dolgozták ki, amelynek az egyszerűbb változatai az APT I a pontvezérlésre, az APT II a két és féldimenziós megmunkálásra alkalmas rendszerek. A MIT APT rendszerét nagy számítógépre írták. Az NC gépek számítógépes programozásának fejlődésében 1965 után az az irányzat alakult ki, hogy az APT alapkoncepcióját megtartva olyan programrendszereket dolgozzanak ki, amelyek az eredeti APT-nél kisebb számítógépeken futtathatók. Ennek a törekvésnek jegyében jött létre az IBM "ADAPT"-ja, az angol National Engineering Laboratory 2P (pontvezérlésű), 2C (pályavezérlés), 2CL (két és féldimenziós megmunkálás) programrendszerei, a német EXAPT1, EXAPT2, a francia IFAPT. Ezek a rendszerek az APT leíró nyelvét és output formátumát kivonatossan ugyan, de megtartották, miközben a programstruktúra belső átalakítása révén a számítógép operatív memóriaigényét csökkentették.

Az itt megemlítettéken kívül számos egyéb nyelv, illetve rendszer készült, amelyeket nem részletezünk. Az APT-hez való hasonlóságuk különböző, egyesek átmenetet képeznek a kézi és a számítógépes programozás között. [91, 37, 72, 73, 122, 104, 63, 89].

Példaképpen az 1. táblázat mutat néhányat az ismertebbek közül.

	NÉV	Kidolgozó intézmény vagy ország
USA	SPLIT CINAP CAMP WALDO AUTOSPOT AUTOMAP PRONTO SYMPAC	Sumststrand Omnimil Cincinnati Westing house Boeing - Wichita IBM IBM GENERAL Electric Univac
Nyugat-Európa	AUTOPIT AUTOPRESS AUTOSURF CLAM COCOMAT MILMAP PAGET PMT2 PROFILEDATA	Pittler IBM (NSZK) Pressed Steel Co.Ltd (Anglia) Olivetti (Olaszország) Howker - Siddeley Aviation (Anglia) Rolls - Royce Ltd. (Anglia) ICT Ltd. (Anglia) Olivetti (Olaszország) ICT Ltd. (Anglia) Ferranti Ltd. (Anglia)
Szocialista országok	JAPO AUTOPROG WTP SAP APROKS SYMAP FORTAP	Csehszlovákia Csehszlovákia Lengyelország Szovjetunió Szovjetunió Német Demokratikus Köztársaság Magyarország

1. táblázat

Az előbbiekben szándékosan kerültük az irodalomban használt APT kompatibilitás kifejezést és használtuk helyette egyszerűen az APT-hez hasonló jelzőt. A kompatibilitás ugyanis az egymás helyett való futtathatóság olyan szigorú követelménye, amelynek a valóságban az APT eredetű nyelvek nem tesznek eleget még akkor sem, ha utasításkészletük többé-kevésbé az APT kivonatának tekinthető.

Ez már előreveti az árnyékát annak, hogy a különböző APT-ből eredő nyelveket kezelő rendszerek megalkotása sokkal több nehézséget okoz, mintha ténylegesen kompatibilis nyelveket kellene közös nevezőre hozni. Ezt bizonyítjuk később az irodalom kritikájában.

4. AZ ALKALMAZOTT TERMINOLÓGIA

Röviden összefoglaljuk az APT típusu rendszerek felépítését és a terminológiát, amelyekre a későbbiek folyamán többször hivatkozunk. Az APT rendszer általános blokksémáját mutatjuk be a 4. ábrán.

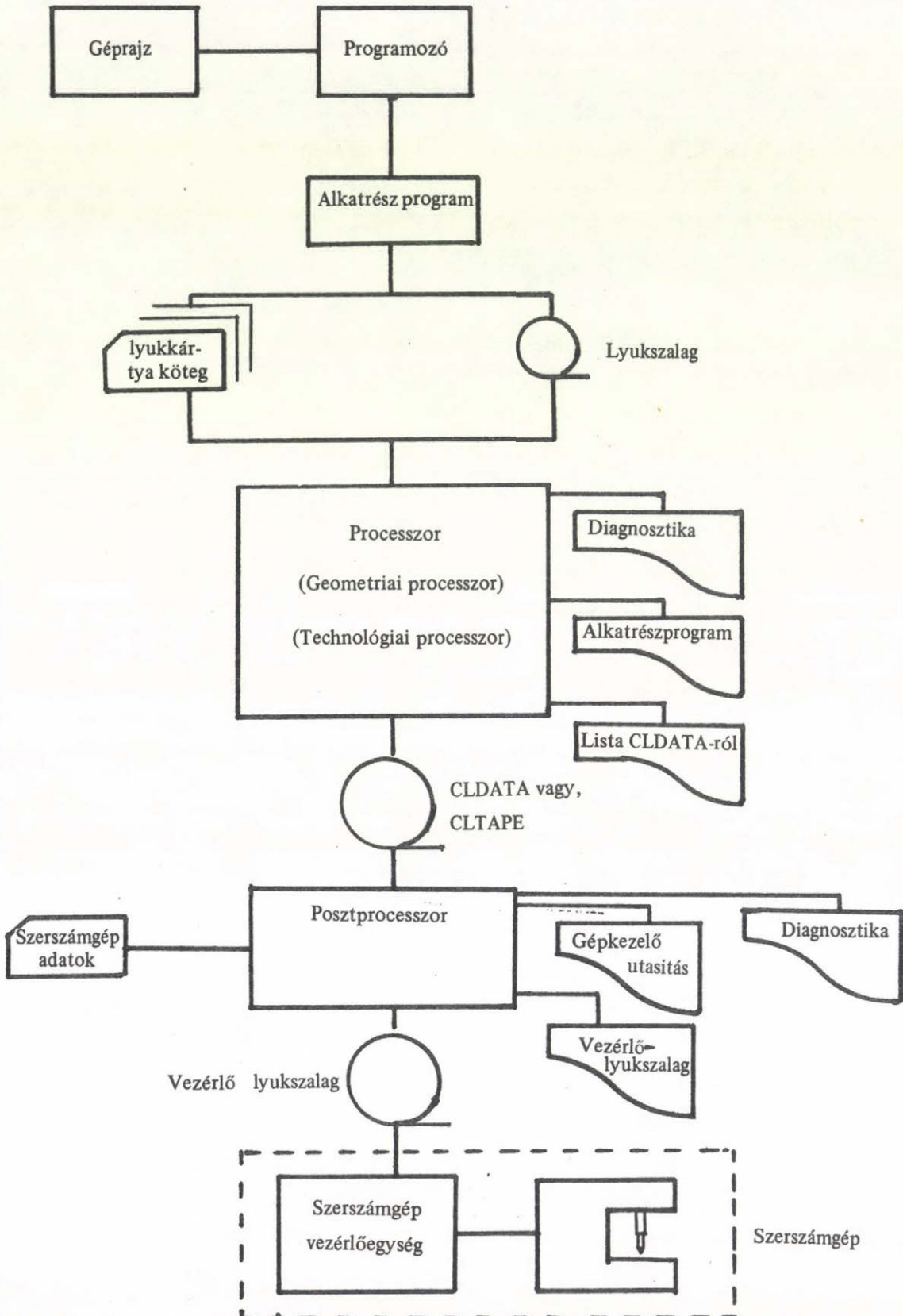
Az APT típusu rendszerek processzorait két nagy csoportra osztjuk. Az első csoportba tartoznak azok, amelyek az alkatrészprogram utasításai közül csak a geometriai és mozgásutasításokat dolgozzák fel. A többi utasítást csak átkódolt formában adják tovább. A második csoportba tartoznak azok, amelyek a technológiai adatokat is feldolgozzák. A technológiai processzor az alkatrészprogram alapján meghatározza a forgácsoláshoz szükséges technológiai adatokat (szerszám kiválasztás, forgácsolási paraméterek, technológiai műveletekhez tartozó mozgásutasítások meghatározása stb.).

A processzor az alkatrészprogramból előállítja az alkatrész megmunkáláshoz szükséges általános vezérlőszalagot.

(CLDATA = Cutter Location Data vagy CLTAPE = Cutter Location Tape). A CLTAPE általános formában tartalmaz minden olyan információt, ami a szerszám gép vezérléséhez szükséges. Adatait az aktuális szerszám gép vezérlőegységének bemenő nyelvére a posztprocesszor transzformálja.

A posztprocesszornak az alábbi feladatokat kell ellátnia:

1. Ellenőriznie kell, hogy az alkatrészprogramban megadott mozgás- vagy egyéb információk az adott szerszámgépen megvalósíthatók-e.



4. ábra

2. Koordináta-értékek átszámítása a szerszámgép koordináta-rendszerébe.
3. Vezérlő utasítások előállítása a szerszámgép vezérlőegységének nyelvén.
4. Szerszámütközések ellenőrzése.
5. Előírás a szerszámtárral rendelkező szerszámgép esetén a szerszámtár feltöltésre.
6. Listák kinyomtatása
 - vezérlő lyukszalagról
 - gépkezelőnek szánt utasításokról
 - diagnózisról.

A programozó a géprajz felhasználásával elkészíti a munkadarab alkatrészprogramját APT típusu nyelven. Az alkatrészprogram tartalmazza a megmunkáláshoz szükséges geometria leírását, szerszámmozgásokat meghatározó információkat, technológiai adatokat stb. Az elkészült alkatrészprogramot a rendszer processzora általában off-line üzemmódban dolgozza fel. A processzor először lefordítja az alkatrészprogram utasításait, szintaktikusan és szemantikusan ellenőrzi azokat. Az alkatrészprogramról, valamint a hibás utasításokról diagnosztikai listát készít. Az alkatrészprogram feldolgozása akkor folytatódik, ha a processzor diagnosztikai rutinja nem talál hibát.

5. AZ APT PROGRAMCSALÁD TAGJAINAK KÖZÖS TULAJDONSÁGAI ÉS AZ EGYES RENDSZEREK ÖSSZEHASONLÍTÁSA AZ IRODALOM ALAPJÁN

Az APT széles körű elterjedése érdekében minden idetartozó programot FORTRAN-ban írtak. Ezzel nagymértékben megkönnyítették azt, hogy sok helyen és különböző számítógép típusokon alkalmazzák, de nem oldották meg a kompatibilitást a szó valódi értelmében, mert ismeretes, hogy többféle FORTRAN változat létezik és, hogy különböző cégek gyártmányaihoz tartozó FORTRAN compilerek sem egyformák. Sokszor egy cég két gépe sem kompatibilis. Annyit azonban elértek, hogy az átírást a FORTRAN egységes alkalmazása megkönnyíti.

A CLDATA rendszer bevezetésével a processzor kimenő nyelvet függetlenítették a szerszámgéptől, legalább is bizonyos határok között.

A posztprocesszorok rendszerét az APT koncepciója alapozta meg. Ezzel csökkentették a rendszer függését a szerszámgéptől. Ezt az elvet az APT minden változatában átvették, de különbözőképpen valósították meg.

A bemenő nyelvet úgy valósították meg, hogy lehetőség szerint egyszerű legyen és könnyen megtanulható. A nemzetközi elterjedés során ez sem érvényesült következetesen, különféle nemzeti elfogultsági okokból (pl. IFAPT).

Minden APT eredetű rendszernek fejlett diagnosztikája van a szintaktikus és szemantikus hibák kiszűrésére. Ezenkívül technológiai ellenőrzést is tartalmaznak.

Az APT-ből leszármaztatott egyéb rendszerek közös tulajdonsága, hogy kisebb terjedelműek, mint az eredeti és,

hogy többnyire csak a két és féldimenziós megmunkálásig mennek el.

A technológiát az APT típusu rendszerek mindegyikébe megosztja szerszámgéptől függő és független részekre, de a megosztás módja rendszerenként eltérő, több vagy kevesebb részt biznak a processzorra.

Az APT CLDATA formátumát lehetőség szerint a belőle le származtatott többi rendszer is megtartotta. Ha ez következetesen sikerült volna, akkor bármely APT típusu rendszerből kikerült CLDATA vezérlőszalag bármely más ilyen rendszer posztprocesszorán futtatható volna. A valóságban azonban csak a hasonlóságig mentek el a valódi kompatilitás helyett. Az APT típusu rendszereket a 2. táblázatban foglaltuk össze.

A különféle APT típusu nyelvek közös vonásai röviden összefoglalva a következők.

Egy alkatrész megmunkálásához a szükséges információkat alkatrészprogramban adják meg. Egy alkatrészprogram mondatokból, (utasításokból, rekordokból) áll, az egyes mondatok szavakból, azok karakterekből vagy számokból tevődnek össze. Felépítésüket szintaktikus szabályok határozzák meg.

A nyelvben használható karakterkészlet az angol abc 32 betője, a 10 arab szám és néhány írásjel (: , . / () \$ % * **+ -). A szavak hossza a számítógép adottságai miatt kötött. Egy szó hat karakternél több nem lehet és számmal nem kezdődhet. A felhasználható szavakat két nagy csoport-

Megnev. típus	Név	Intézmény neve	Ország	Aut.mély-sége	Minimális számítógép igény	Megfogalm. nyelve	Irodalom	Megjegyzés
3 Dim.	APT III.	MIT	US	G	IBM 360/40	FORTRAN	138	
2 1/2 Dim. Maró megm.	ADAPT	IBM	US	G	IBM 360	FORTRAN	91,153,26 , 169,137	Utasítás készlete majdnem az APT III-al azonos
	NELAPT 2 CL	NEL NEL	Anglia Anglia	G.T. G	32 K szó 24 bit/szó	FORTRAN FORTRAN	132 140	Szótári szavak tömör formáit megengedik GOZIG, GORND
	EXAPT 3	EXAPT Verein	NSZK	G.T.	64 K 64 bit/szó	FORTRAN	120,107,117, 54,95	Nem valósult meg
	IFAPT	ADEPA	Francia	G	16 K 56 bit/szó	FORTRAN	141,129	
2 Dim. Eszterga	APT II.	MIT	US	G	IBM 360	FORTRAN		
	AUTOMAP 2 C	IBM NEL	US Anglia	G G	IBM 360 32 K szó 24 bit/szó	Assembl. FORTRAN	153 115,60,80	Szótári szavak tömör formái
	EXAPT 2	EXAPT Verein	NSZK	G.T.	64 K 64 bit/szó	FORTRAN	151,54,117,95	
	IFAPTC	ADEPA	Francia	G	16 K 56 bit/szó	FORTRAN	142,129	
1 Dim. Furó megm.	APT I.	MIT	US	G	IBM 360	FORTRAN		
	AUTOSPOT	IBM	US	G	IBM 360	Assembl.	153,168	Pontmintázattal bővíti az APT III-at
	2 PL	NEL	Anglia	G	32 K szó 24 bit/szó	FORTRAN	115,60,80	Szótári szavak tömör formái
	EXAPT 1 IFAPTP	EXAPT Verein ADEPA	NSZK Francia	G.T. G	32 K szó 32 bit/szó 16 K 56 bit/szó	FORTRAN FORTRAN	150,74,117,95 143,129	

G Geometria
T Technológia

2. táblázat

ra osztjuk. Az első csoportba azokat a szavakat soroljuk, amelyeket a programozó ad meg különböző utasítások megszervezésére. Ezeket a szavakat azonosítóknak vagy szimbólumoknak nevezzük. A nyelvben szereplő szavak második csoportjába soroljuk a kötött vagy szótári szavakat. A szótári szavak angol szavak rövidítéséből vagy összevonásából alakultak ki. Formájuk és értelmük kötött. Számuk az APT típusu nyelveken belül is változik. (Például a 2CL nyelv 177 db, az EXAPT1 nyelv 105 db szótári szót tartalmaz). A kötött szótári szavakat szintén két csoportra osztjuk mondatbeli szerepüktől függően. Az első csoportba tartoznak az ugynevezett főszavak, amelyek az utasítások típusát határozzák meg. A második csoportba tartoznak a módosítók, amelyek a főszó által meghatározott utasítást teszik egyértelművé.

Az APT típusu nyelv mondatai felépítésük szerint két csoportot alkotnak aszerint, hogy az alkatrészprogramban definíciós vagy végrehajtó szerepet töltenek be.

A definíciós jellegű utasítások általános formája a következő:

$$S = F/A, M, S_n$$

ahol az

S szimbólum vagy azonosító

F főszó

A adatok

M módosítók

S_n előzőleg az alkatrészprogramban már meghatározott definíciós utasítás azonosítója.

Az A, M, S_n közül bármelyik elmaradhat.

A programozó a definíciós utasításokkal geometriai alakzatokat, technológiai előírásokat, koordináta-rendszer transzformációkat határoz meg. A definíciós utasítások azonosítók segítségével beépíthetők újabb definiáló vagy végrehajtó rutinokba.

A végrehajtó utasítások általános alakja:

F / A, M, Sn

Az egyes rekordokat vagy felhasználásuk vagy az alkatrészprogramban betöltött szerepük szerint csoportosíthatjuk. A geometriai alakzatokat geometriai utasításokkal, a szerszám mozgását mozgásutasításokkal, a megmunkáláshoz szükséges technológiai előírásokat a technológiai utasításokkal stb. adják meg.

Néhány kötöttségtől eltekintve, a rekordok sorrendje tetszőleges lehet. Összefoglaljuk a kötöttségeket:

1. Az alkatrészprogramot mindig a PARTNO utasítással kell kezdeni.
2. Az alkatrészprogram a FINI utasítással fejeződik be.
3. Az alkatrészprogramban mindig meg kell adni a MACHIN, vagy a NOPOST utasítást (szerszámgép megnevezése). Néha több MACHIN utasítás is szerepelhet egy programban.
4. A végrehajtó utasítások sorrendjét az általuk aktivizált műveletek sorrendje határozza meg. Az egyes rekordokat a processzor hívásuk sorrendjében dolgozza fel, ezért a funkcionális műveletek hívásuk sorrendjében aktivizálódnak.
5. Az egyes utasításokban, csak olyan más utasításra hivatkozhatunk azonosítóikkal, amelyeket már az alkatrészprogramban előzőleg meghatároztunk.
6. Egy névvel csak egy utasítás nevezhető meg.

A fenti kötöttségek a gyakorlatban kialakítottak egy programozási sorrendet:

- Az alkatrészprogram megnyitása (PARTNO)
- Postprocesszor megnevezése (MACHIN)
- A processzor számításaihoz szükséges adatok megadása
- Az alkatrész geometriájának leírása
- Technológia definiálása
- A megmunkáláshoz szükséges kezdő értékek beállítása
- Technológiai, valamint a szerszám mozgását aktivizáló utasítások leírása
- Alkatrészprogram befejezése (FINI).

A leírt sorrend betartása nem kötelező, de több előnye van. Például az alkatrészprogram áttekinthetőbbé, a hibák megtalálása egyszerűbbé válik stb.

6. AZ INTERAKTIVITÁS SZEREPE

Az ember-számítógép kölcsönös, kölcsönös együttműködésével, az interaktív kapcsolat gépészeti tervezési és technológiai felhasználásával és annak kialakításával a külföldi és hazai irodalom részletesen foglalkozott. [81, 65, 25, 80, 82, 121, 124, 125, 112, 24, 19, 42, 67, 113, 69, 71, 1, 70, 22, 92, 144, 133, 46, 45, 31, 134, 55]. Megemlítjük az MTA Számítástechnikai és Automatizálási Kutató Intézete Közleményei 1969. évi 5. számát és az OMFB 1972. évi jelentését, amelyek részletes áttekintést adnak [38, 171]. A CAD (Computer Aided Design) és interaktív tervező rendszerek kutatásával az MTA SzTAKI 1969 óta foglalkozik [3, 55, 39, 40].

Az interaktív számítógépek NC programozó rendszereiben eleinte az interaktivitás perifériája konzolirógép volt és célja az alkatrészprogram diagnosztizálása [156, 154, 35, 79, 157]. Az első katódsugárcsöves interaktív rendszer a Conversational APT = (CONAPT) és a SPLIT [61] volt. Európában pedig a NATIONAL ENGINEERING LABORATORY (NEL) fejlesztette ki a Conversational 2CL interaktív alkatrészprogramíró rendszert. [79, 112]. Tekintettel arra, hogy ez a rendszer közel áll a dolgozatban kifejtett témához, részletesen ismertetjük.

Az operátor a display képernyő előtt ülve a számítógéppel való társalgás közben írja az alkatrészprogram utasításait. A programozónak lehetősége van arra, hogy társalgás nélkül on-line írja az alkatrészprogram mondatait (segítséget nem kér a rendszertől). Az operátor ebben az üzemmódban is visszajelzést kap, ha hibás utasítás kerül a pro-

cesszorba, de a javítást csak akkor tudja elvégezni, ha a "Conversational" programcsomagot újra behívja a számítógépbe. Erre az alkatrészprogram írása közben lehetőség van.

A "Conversational" üzemmód háromféle üzenettípussal dolgozik. Az első típusu üzenet tartalmazza annak a fő programrésznek a nevét, amelybe a program került, pl:

****EDITING****, ****ERRORS**** vagy ****PROCESSING****. Az e típusba tartozó üzenetek két-két csillag között kerülnek kiírásra. A második típusba tartozó üzenetek a diagnosztika eredményét közlő kiírások (a megírt utasítások visszaigazolások, vagy hibakijelzések stb.). A harmadik típusu üzenetre a rendszer az operátortól minden esetben választ vár. Ilyen üzenet például a ***NOW***, amelyre új utasítást, a ***CHOOSE***, amelyre valamilyen manipulációs eljárást kér. Ezek a következők:

LIST/ALL	az eddig megírt utasítások listázása
LIST/SOME	tetszőleges utasítások vagy utasítás kiírása
INSERT	meghatározott helyre új utasítás behelyezése
DELETE	meghatározott helyről utasítások törlése
CHANGE	utasítások felcserélése
PUNCH	az egész alkatrészprogramról lyukszalag készítése.

A választ váró utasítások egy-egy csillag közé kerülnek.

Az operátor "Conversational" üzemmódban minden új ***NOW*** kérésre egy új utasítást ír le. Az utasítás, miután keresztülment a diagnosztika vizsgálaton, visszaigazolásként egy újabb ***NOW*** kérés íródik ki a display képernyőre. Hibás utasítás esetén a számítógép közli a hiba típusát.

sát és a hibás kártya sorszámát. A hibajelzés után megjelenik újra a *NOW* felirat, amely kérésre az operátor megadja a helyes utasítást, vagy valamilyen manipulációs lehetőséggel kijavítja a hibát, vagy módosítja a programot. Az alkatrészprogram befejezésekor a szélesnyomtatón kiiródik a szintaktikusan hibátlan alkatrészprogram. Kiiródik továbbá az alkatrészprogramban szereplő utasítások hatására minden olyan számított érték, amelyet a programozó kért. Végül a szerszámgépet vezérlő lyukszalag tartalma nyomtatódik ki.

Ez és még néhány hasonló vagy megegyező, de különböző elnevezésű rendszer nem igényel katódsugárcsőves display-t, hanem csak konzolirógépet.

Viszont a Lockheed Missiles & Space Co. 1967-ben megjelent tanulmánya már olyan rendszerről számol be, amely katódsugárcsőves grafikus display-t használ. [146] A tanulmány azért figyelemre méltó, mert azt fejtegeti, hogy a CAD (Computer Aided Design) és az NC (Numerical Control) nem választhatók el egymástól, mindezek pedig nem valósíthatók meg a CG (Computer Graphics) nélkül, legalább is gazdaságosan nem. Ez még akkor is igaz, ha a hozzá szükséges berendezések nem olcsók, és ezek különösen nem voltak azok a cikk írása idején, 1967-ben.

A terminál hardware berendezése grafikus display, fénycecuza, funkcionális és alfanumerikus tasztatura. A megmunkálendő alkatrész geometriáját két módon lehet a képernyőn megjeleníttetni. Az egyik, hogy lyukkártyákon vagy lyukszalagon adják be a szükséges geometriai információkat. Ez lehetséges APT nyelven is. A másik módszer, hogy az operátor rajzolja a képernyőre a kívánt ábrát. A funkcionális billentyűzet segítségével kell kiválasztani azt

a geometria tipust (PARALLEL, CIRCLE, POINT, LINE THRU POINT AT DIRECTION, LINE POINT TO POINT, FILLET.TRANSFORMATION), amelyet a képernyőn meg akarunk határozni. A pontos geometriai alakzat meghatározásához még a rendszer által megadott menülistákra, illetve kérdésekre kell válaszolni fényceruza, funkcionális vagy alfanumerikus tasztatura segítségével. A szerszámmozgások meghatározásánál először szintén a funkcionális tasztatura segítségével kell kiválasztani azt a mozgásfajtát, amelyet alkalmazni akarnak. Ezután vagy a fényceruza, vagy az alfanumerikus tasztatura segítségével kell válaszolni a rendszer által feltett kérdésekre. Az alkatrészprogramban szereplő egyéb, a szerszámgépre, illetve a számítógép számításaira vonatkozó paramétereket hasonlóképpen adják meg. - A rendszer biztosít különböző módosítási lehetőségeket is. Az operátor a program írása közben módosíthatja akár a képernyőn megjelenített rajzot (geometriát), akár a már meghatározott egyéb adatokat is. A rendszer az elkészített alkatrészprogramból hibátlan vezérlő lyukszalagot állít elő.

Egy másik hasonló típusu rendszert 1969-ben az IBM mutatott be a párizsi nemzetközi szerszámgép-kiállításon. [76] A grafikus interaktív alkatrészprogramíró rendszer IBM/360/65 típusu számítógépet használ IBM 2250 típusu grafikus display terminállal. A display-hez tartozó perifériák segítségével tetszőleges ábra rajzolható fel a képernyőre, illetve a felrajzolt ábra módosítható a feladatnak megfelelően. Az ábrát interaktív módon rajzolják fel a képernyőre. A gépészetben használt geometriai alakzatokat hat részre osztották fel (pontok, egyenesek, körök, körívek, kupszeletek, pontok által meghatározott görbék), amelyek közül az operátor kiválaszthatja, hogy melyik alakzatot akarja meghatározni. Ha egy kiválasztott geometriai alakzat meghatározásának több módja van, egy újabb menüvel

kell kiválasztani a feladathoz legjobban alkalmazkodó megoldást. A menüből való választással az operátor egy szubrutin csomagot aktivizál, amely a kívánt feladatot megoldja. A kiválasztott szubrutin csomag a számára szükséges paraméterek beolvasása után meghatározza a kívánt geometriai alakzatot, az a képernyőn megjelenik, valamint tárolódik egy későbbi feldolgozás céljára. - A rendszerrel pont- és pályavezérlésű megmunkálásokhoz vezérlő lyukszalagot lehet készíteni. A megmunkáláshoz szükséges geometriai adatokat az ismertetett módon kell felrajzolni a display képernyőre. A szerszám mozgásának, illetve a megmunkálás sorrendjének meghatározásához szintén menüt használ a rendszer. Az operátor fényceruza segítségével a menüből választja ki a szükséges mozgástípust. A megmunkáláshoz szükséges numerikus értékeket, a posztprocesszor kiválasztását, valamint egyéb információkat a programozó adja meg, válaszolva a rendszer által feltett kérdésekre. Az operátor az alkatrészprogram meghatározása után a kiválasztott szerszámgéphez tartozó vezérlő lyukszalagot kapja meg, amely szalag plotteren automatikusan ellenőrizhető.

Az egyik legismertebb és legjobban elterjedt a CDC (Control Data Corporation)-nek az APT/IG S (Automatically Programmed Tools/Interactive Graphics System) rendszere [27]. Létrehozásával az alábbi feladatok megoldását kívánták elérni.

- Kézi alkatrészprogramírás megszüntetése.
- APT alkatrészprogram lyukasztásának megszüntetése (nem kötelezően).
- Képernyőre off-line rajzolás megszüntetése.
- Alkatrész legyártásához próba megmunkálások számának csökkentése.

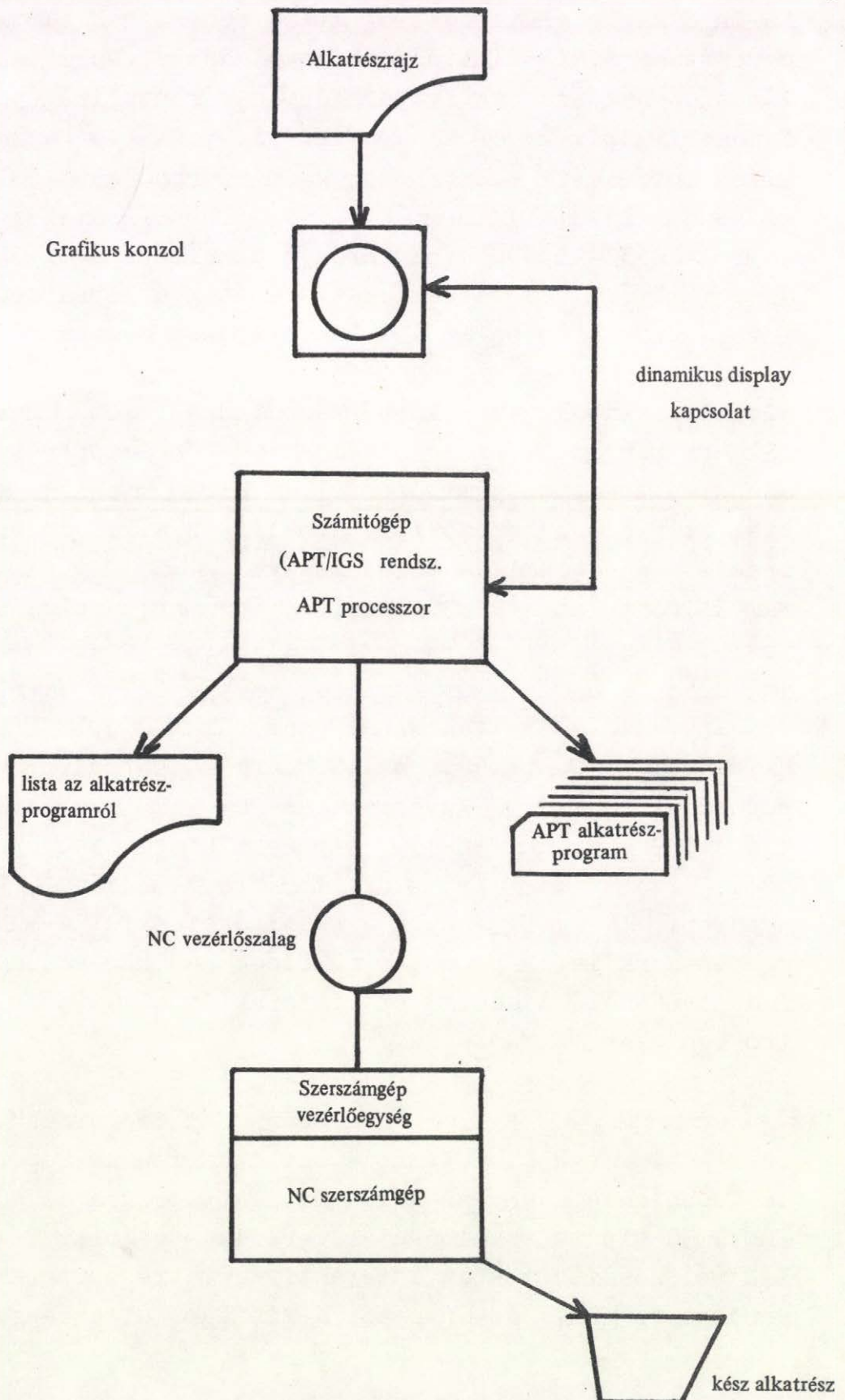
- Alkatrészprogramozók hatékonyságának növelése.
- NC műhelyek hatékonyságának növelése.
- Számítógép futtatási idejének csökkentése.
- Egyszerű és gyors alkatrészprogram-módosítás lehetőségének megteremtése.
- Optimális alkatrészprogram elkészítése.

A rendszert CDC 6000 sorozatu számítógépekre dolgozták ki CDC 274 grafikus display-vel. Blokksémáját az 5. ábrán mutatjuk be, amely jól szemlélteti a display és a számítógép közti kapcsolatot.

Az alkatrészprogramhoz szükséges összes információk megadása (geometria, technológia, mozgás) interaktív formában történik. A munka befejezése után a rendszer az NC vezérlő lyukszalagon kívül az APT alkatrészprogram lyukkártya csomagját is kiadja.

Kevésbé fejlett, de olcsóbb grafikus terminál alkalmazásával dolgozták ki a Computer - Aided Design Centernél a GNC (Graphical NC processor) rendszert. [30] Az alkatrész megmunkálásához szükséges geometriai adatokat a GNC rendszerhez tartozó geometriai nyelv segítségével off-line kell a display képernyőre felvinni. A technológiai adatok, illetve a szerszám mozgásának meghatározása interaktív formában történik.

A Német Szövetségi Köztársaságban az Aacheni Műszaki Főiskolán és az EXAPT Vereinnél dolgoztak ki interaktív grafikus NC alkatrészprogramíró rendszert, amelyet 1972-ben fejeztek be [116, 131]. Integrált tervező és gyártó rendszerről 1973-ban számoltak be a Prolamat'73 Konferencián [77].



5. ábra

Az interaktív alkatrészprogramíró rendszerek harmadik csoportjába a mini-számítógépes megoldásokat soroltuk. Ezek a rendszerek interaktív perifériának konzolirőgépeket, alfanumerikus, vagy olcsó grafikus display-eket használnak. Ilyen interaktív alkatrészprogramíró rendszert dolgoztak ki Houdaille Industriesnél [160] PDP-8/L mini-számítógépre ASR-33 típusu konzolirőgép alkalmazásával. A rendszer működése a következő: az operátor a géprajzról vett információk alapján az ASR konzolirőgép segítségével írja az alkatrészprogram mondatait. A számítógépbe adott utasítások, szintaktikus ellenőrzés után, egy közös memória file-ra kerülnek. Hibás utasítás esetén az operátor visszajelzést kap a számítógéptől. A hibás utasítás kijavítása után a programozó folytathatja az alkatrészprogram írását. A programírás befejezésével a teljes alkatrészprogram kiíródik az ASR-33 konzolirőgépen egy utolsó ellenőrzésre. Ha az operátor a megírt alkatrészprogramot elfogadja, arról lyukszalagot kérhet. A Japán OKI Electric Ind. Co. Ltd.-nél dolgozták ki, alfanumerikus display felhasználásával, a "Conversational MINIAPT" alkatrészprogramíró rendszert a MINIAPT programozására. A "Conversational MINIAPT"-et az OKITAC-4500 típusu mini-számítógépre (20 K szó, 1 szó 16 bit) írták diszk háttérmemóriával, FORTRAN nyelven. [162]. Ugyanezen a számítógépen fut a MINIAPT processzora is. A rendszer működése hasonló a Houdaille Industriesnél kiadolgozott interaktív alkatrészprogramíró rendszer működéséhez.

Az e csoportban ismertetett rendszereknél fejlettebb rendszert dolgoztak ki a francia Ecole Nationale Supérieure de Techniques Avancées-nél a Minifapt programozására. IBM 1130 típusu számítógép és grafikus display felhasználásával készítették az interaktív alkatrészprogramíró rendszert. [28]. Az operátor a Minifapt alkatrészprogra-

mot a display-hez tartozó alfanumerikus tasztatura segítségével készíti el. Az alkatrészprogram elkészítése után az operátor a display-n megjelenített ábrán korrigálhat és a megmunkáláshoz szükséges egyéb információkat dialógus formában adja meg. Az elkészített hibátlan alkatrészprogramot ugyanezen a számítógépen futó Minifapt proceszszora dolgozza fel.

Alkatrészprogram készítésre mini-számítógépet egy másik konfigurációban is szoktak használni. Californiában a Tridea Electronics az ALTAPE rendszert dolgozta ki [154], amelyben a mini-számítógépet egy rajzletapogató berendezéssel kapcsolták össze. A rajzletapogató fej a rajzot letapogatja. A letapogatott információ jelátalakítón keresztül kerül a számítógépbe, ahol egy program összeállítja az alkatrészprogramot. Az operátor egy konzolirógép segítségével avatkozhat be, illetve adhatja meg az egyéb információkat. A felhasznált számítógép Varian Data 620/i típusu.

7. SPECIÁLIS HAZAI SZEMPONTOK

A KGST országokban kifejlesztett NC számítógépes programozó rendszerek egymástól függetlenül jöttek létre. E munkák összefogása és egyesítése érdekében a KGST Gépipari Állandó Bizottsága döntést hozott egy közös NC automatikus programozó rendszer létrehozására. Az e munkában való részvételre Magyarországon létrejött a Szerszámgép Programozási Egyesülés (SPE) KGST munkacsoportja, amely hazánk képviselőjét koordinálja. 1970. szeptember 18-án a KGST Gépipari Állandó Bizottsága jóváhagyta azt a magyar javaslatot, hogy a processzort a posztprocesszorral összekötő közbenső nyelv az ISO /TC 97/SC5 javaslata szerinti CLDATA legyen. A beterjesztett javaslatot a Gépipari Állandó Bizottság 1971-ben Kecskeméten tartott ülésén elfogadta és Magyarországot bizta meg "A gépi programozás egységes rendszerének posztprocesszorai tervezet" kidolgozásával, a közbenső nyelv alkalmazás kísérleti metodikájának elkészítésével stb.

Az egységes programozó rendszer létrehozásánál figyelembe veszik az idevonatkozó ISO javaslatokat, ami értelmezésük szerint azt jelenti, hogy a tervezett rendszer is APT jellegűvé válik.

8. A VONATKOZÓ IRODALOM KRITIKÁJA

8.1 Az APT rendszerek és nyelvek kritikája

Az APT típusu programozási rendszerek elterjedésének nehézségei voltak, illetve vannak még ma is. Néhány súlyos probléma merült fel. Az akadályozó tényezők gyökere a számítógép. Annak ellenére, hogy az APT III. szegmentálásával sikerült a memóriaigényt lejjebb szorítani, az egyszerűbb rendszerek is csak közepes nagyságu számítógépeken futtathatók. Nagyon sok kis, közepes vállalat nem tudott megfelelő nagyságu számítógépet biztosítani. A helyzetet nehezítette az a körülmény is, hogy az NC szerszámgépek számának növekedésével a vállalatok szerszámgép-parkja természetesen nem maradt homogén. Így egy üzemet általában nem elégít ki egy vagy két automatikus rendszer, többre van szüksége. Azoknak az üzemeknek pedig, amelyek rendelkeztek megfelelő nagyságu számítógépekkel, ugyancsak komoly problémákkal kellett megbirkóznuk. Újabb szakember gárdát kellett kinevelniük, akik az automatikus programozási rendszerek állandó karbantartásáról gondoskodtak.

A problémák felszámolására két megoldás született. Az egyik az volt, hogy az NC szerszámgépek automatikus programozó rendszereinek kezelésére, karbantartására és az alkatrészprogramok futtatására szervizek, egyesülések, vállalatok jöttek létre. (Ilyen szervizek például az Auco Aerostructures Corp. [108], az Equipment Group TRW Inc. Ohio-ban [43], a Computer Service Company, a Numerical Control Information Utility Philadelphiában [21], Európában a BARIC, az NCS, az EXAPT Verein stb.)

Ezek a szervizek nagy számítógépekkel rendelkeznek, többfajta automatikus programozó rendszert kezelnek a felhasználók szerszámgépeinek posztprocesszoraival együtt. A futtatások, illetve az egyéb szolgáltatások díjazása ellenében gondoskodnak az egyes cégek alkatrészprogramjainak feldolgozásáról. - Ez a megoldás a problémáknak csak egy részét oldotta meg. Újabb problémákat okozott az alkatrészprogramok készítése. Bár az automatikus rendszerek kidolgozásánál az volt az egyik cél, hogy az alkatrészprogramok elkészítéséhez ne legyen szükség programozó szakemberre, ezt nem sikerült megvalósítani. Az eredeti elképzelés szerint a programozási munkára gépészmérnökök vagy átképzett szakmunkások is megfeleltek volna az automatikus rendszerek és a számítógépek elemi ismerete nélkül is. Sajnos ezt az elképzelést a gyakorlat megcáfolta. M. Lester Rizon szerint [108] az alkatrészprogramok elkészítéséhez komoly műszaki felkészültségű emberek szükségesek. Azt ajánlja, hogy az alkatrészprogramok elkészítésére programozó csoportokat hozzanak létre, amelyeknek élére technológusokat kell állítani. A vezető technológusnak nagyon jól kell ismernie az üzemet, annak minden NC szerszámgépét, illetve az azokhoz szükséges technológiát, valamint azokat az automatikus programozó rendszereket, amelyeket az üzem használ. Mielőtt egy alkatrészprogram írása elkezdődne, az alkatrészről, illetve annak technológiájáról, valamint a programozási utmutatásokról a vezető technológusnak művelettervet kell készítenie. A műveletterv alapján alacsonyabb képességű programozó is el tudja készíteni az alkatrészprogramot.

A gyakorlatban még a nagyon gondosan elkészített alkatrészprogramokba is kerülnek hibák (elírás, lyukas-

tási hiba stb.), amelyek csak a számítógépen való futás közben derülnek ki. Egy alkatrészprogram teljes kijavításához 3-5-szöri futás szükséges. A szervizekben futtatott alkatrészprogramoknak 3-5-ször kell fordulniuk a szerviz és az üzem között. Sokszor egy alkatrészprogram hibátlan futtatásához egy hét is szükséges [108, 79, 43]. Egyes szervizek az alkatrészprogramok megírását is vállalják megfelelő rajzok alapján. Ezek a megoldások sem váltak be, mivel az alkatrészprogramozók a helyi adottságok ismerete nélkül készítették a programokat.

H. Burton [10] szerint a fenti módszerek a különböző hiányosságaik ellenére kb. a felére csökkentették a programozási költségeket.

A másik megoldás, amely a számítógéppel nem rendelkező vállalatok problémáin igyekszik segíteni, az időosztásos (time-sharing) rendszerek alkalmazása, távadat feldolgozó (TAF) hálózaton át. Az üzemek egy végállomás (terminál) bérlésével a számítógépek közvetlen közelébe kerülnek. A programozó a végállomáshoz tartozó perifériális berendezéseken keresztül kommunikál a számítógéppel, legegyszerűbb esetben konzol-írógépen. A General Electric Co.-nél állítottak fel ilyen rendszert először [53]. A felhasználók számára előnyös, hogy nagy teljesítményű számítógépekkel kerülnek közvetlen kapcsolatba, magasabb szintű programozó rendszereket is igénybe tudnak venni. Ezeken a számítógépeken az alkatrészprogram átfutási ideje rövid. Az esetleges programozási hibák a terminálon észlelhetők, így a programozó azonnal javítani tud.

[90]

A time-sharing gépekre csatlakozó TAF rendszerek újabb fejlődést indítottak el azzal, hogy lehetővé tették katódsugárcsőves megjelenítő eszközökkel felszerelt végállomások bekapcsolását az interaktív alkatrészprogramíró rendszerekbe. Az automatikus NC programozás problémáit ezek a rendszerek sem tudták teljesen megoldani. A felmerült újabb problémákat az alábbiakban foglaljuk össze:

1. Igaz, hogy az alkatrészprogram készítő szervizek és a time-sharing rendszerek nagy lehetőségeket kínáltak fel a felhasználóknak, de az alkatrészprogramok készítése, illetve futtatása költségesnek bizonyult. Az alkatrészek gépi programozása nem mindig gazdaságos ilyen körülmények között. Az egyszerűbb furó megmunkálások esetében pl. a geometriai és a mozgásutasítások olyan egyszerűek, hogy kézi programozással olcsóbban és rövidebben célt lehet érni. Még inkább igaz ez, ha csak olyan rendszerek állnak rendelkezésre, amelyekben a technológia automatizálása sem megoldott.

E hézagok pótlására egyes cégek saját céljaikra különböző rendszereket dolgoztak ki. Ilyen például az Olivetti [86] rendszere, amely time-sharing felhasználásával a furó megmunkálások leírására szolgál. A program a geometriai meghatározáson kívül magas színvonalu technológiai automatizálást valósít meg, de ki- és bemenő nyelve nem kompatibilis az APT-vel.

2. Az alkatrészprogramban lévő hibák a számítógép futtatási költségeit nagyon megnövelik. A processzorok az egyszerűbb szintaktikus hibákat

már a futás elején kiszűrjük, de vannak hibák, amelyek csak a posztprocesszorban jelentkeznek. Minden egyes próbafutásnál az egész processzort igénybe kell venni, ami azt jelenti, hogy az amugy is költséges gépidőt duplájára vagy háromszorosára kell növelni. A hazai tapasztalatok is ezt bizonyították. Az EXAPT1 INTRAN programszegmense, amely beolvasó és diagnosztizáló feladatokat lát el, az egész processzornak csak egy kis része. Egy-egy alkatrészprogram belövési periódusában az INTRAN programszegmens futási ideje (több ismételt futás miatt) hosszabb volt, mint az egész processzor futási ideje.

Time-sharing üzemmódban ez a probléma fokozottan jelentkezik a drágább számítógépidő miatt. Hibátlan alkatrészprogramok előállítására több módszert dolgoztak ki. Egyik, hogy a kész lelyukasztott alkatrészprogramot a processzoron való átfutás előtt mini-számítógépen szintaktikus ellenőrzésnek vetik alá [160] vagy egy pre-processzorral szűrjük ki a hibákat. [110]. Hibátlan alkatrészprogramokat lehet írni az interaktív alkatrészprogramíró rendszerek alkalmazásával.

3. Több felhasználó bírálata alá veszi az APT típusú rendszereket time-sharing felhasználás szempontjából. Bernard-Feinberg [35] szerint az APT és az ADAPT rendszerek azért nem alkalmasak time-sharing alkalmazásra, mert utasításkészleteik túl nagyok és az egyes utasítások túl redundánsak. E fő indok vezette a Manufacturing Data Systems Inc.-et, hogy kidolgozza a Compact II. két, három és négy tengelyű megmunkálások

automatikus programozási rendszerét time-sharingre. [35]. Bemenő nyelve szabad formátumu, utasításai angol szavakból állnak (nem kompatibilis az APT-vel). Az alkatrészprogramok írására interaktív szervezést is megvalósítottak. Philippe Goudissart cikkében [41] szintén a processzor túl nagy memóriaigényét, valamint a bemenő nyelv gazdaságtalanságát bírálja. Megemlíti, hogy az APT rendszerek mind a kötött szótári szavakra, mind a megengedett szimbolumok használatára hat karaktert engednek meg. A hat karakter használata egyedül a 36 bites szóhosszuságú számítógépeknél gazdaságos. A byte strukturájú számítógépek általában négy byteszavasak, ami négy karakternek felel meg, tehát dupla szóhosszal kell dolgozni. Bírálja továbbá azt is, hogy a processzorok (geometria, technológia) nincsenek eléggé szegmentálva és így feleslegesen nagy memóriarészt kötnek le. A fenti okok miatt dolgozta át a General Electric az APT nyelvet REMAPT (REMOte Adapt) néven time-sharing rendszerre [53]. Hasonló okok vezették a Computer Sharing Inc.-et, hogy kifejlessze az ADAPT-PLUS rendszert, amely az ADAPT rendszert teszi alkalmassá time-sharing üzemmódra [156].

Az 1960-as évek végén a mini-számítógépek elterjedésével az automatikus NC programozás területén új törekvések figyelhetők meg. Az említett programok az automatikus NC rendszerek elterjedését akadályozó okokat szerették volna megszüntetni. Elsősorban a nagykapacitású számítógép kiküszöbölését, a processzorok nehézkes

kezelését, az alkatrészprogramok készítésénél mutatkozó nehézségeket, és nem utolsó sorban az alkatrészprogramok házon belüli elkészítését remélték megoldani. Már az első eredmények is biztatóak voltak, mivel a mini-számítógépek nem igényeltek nagy beruházási költséget és a futási költségük is minimális volt.

A kifejlesztett mini-számítógépes rendszerek közül csak néhányat említünk meg.

A megmunkálási feladatok között vannak olyanok, amelyeknek automatikus programozása APT típusu programmal nem gazdaságos. Ezt a problémát akarta megoldani a PERA (Production Engineering Research Association) által létrehozott mini-számítógépes rendszer, amely a kézi programozási időt 80 %-kal csökkenti. A rendszer bemenő nyelvét az APT szókészletéből kiválasztott 40 szó tartalmazza. A FORTRAN nyelven megfogalmazott rendszer 8 K(24 bites szó)operatív memóriás számítógépet igényel [159].

Californiában a Pacific Data Systems Inc. egy kisszámítógépes rendszert dolgozott ki a PDS 1020-as számítógépre 4 K szó operatív memóriával. A SHOP (SHop Oriented Program) rendszer mind a pont, mind az egyszerűbb pályavezérlésű NC szerszámgépek automatikus programozására alkalmas. Bemenete assembler szintű nyelv. Az alkatrészprogramok elkészítésére nyomtatványokat használnak, amelyet egy műhelybeállítottságu programozó könnyen ki tud tölteni. A rendszer négy részből áll: a bemenő nyelv, a geometriai program, szerszám gép számára szervező program (posztprocesszor) és listázó program.

Nem kompatibilis az APT-vel. [49]

A Californian United Computing Inc. UNIAPT rendszere teljesen APT kompatibilis, funkcióiban is - csekély kivétellel - ugyanazt képes teljesíteni, mint az APT III. Az UNIAPT futtatható PDP-8, IBM 1130, valamint GE-PAC 4020 típusu mini-számítógépeken 64 K lemezmemóriával kiegészítve overlay szervezésben. [159, 96].

Ebbe a csoportba kell még sorolnunk a már említett legelterjedtebb APT-kompatibilis mini-számítógépes MINIAPT [166] és a MINIFAPT [28] rendszereket is.

8.2 Az interaktív rendszerek kritikája

Az interaktív alkatrészprogramíró rendszerek fejlesztésére az ipart és a kutatást az alkatrészprogramok elkészítésének nehézségei és az alkatrészprogramok magas futtatási költségei ösztönözték. A fejlődést meggyorsította a time-sharing rendszerek elterjedése. A költségek csökkentése megkövetelte, hogy az alkatrészprogram meddő futásainak számát minimálisra csökkentsék. Ez indította a rendszertervezőket arra, hogy olyan interaktív rendszereket dolgozzanak ki, amelyek segítségével hibátlan alkatrészprogram készíthető.

Az interaktív alkatrészprogramíró rendszerek létrehozásának a célja az volt, hogy minél rövidebb idő alatt szintaktikusan helyes és szemantikus hibáktól nagyrészt mentes alkatrészprogramokat írjanak. Az APT típusu rendszerek processzorai az alkatrészprogramot mondatról-mondatra olvassák be. Minden mondat beolvasását követi az utasítás szintaktikus ellenőrzése. Hibás

utasítás esetén a processzor kiírja a hiba típusát és a hiba jellegétől függően beolvassa a következő mondatot, vagy megszakítja a program futtatását. Mivel a processzorok mindazt a diagnosztikai vizsgálatot elvégzik, amit az interaktív alkatrészprogramíró rendszerrel akarnak elvégeztetni, az alfanumerikus display-t alkalmazó rendszerek zömét úgy valósították meg, hogy az eredeti processzorokat interaktív programrészekkel egészítették ki. Igaz, hogy a rendszerek létrehozási költségei lényegesen csökkentek, de a futtatási, azaz alkalmazási költségeik nem mondhatók optimálisnak. Az alkatrészprogram írásához ugyanis az egész processzort be kell olvasni a számítógépbe, ami továbbra is nagy számítógépet igényel. A processzorokban elhelyezett diagnosztikai programok sorrendje erre a célra nem optimális.

Az interaktív, vagy "conversation" elven működő rendszerekkel az alkatrészprogramírás idejét lényegesen csökkentették. Lerövidült a program írásának ideje azért, mert, ha a programozó valamilyen hibát követett el, a diagnosztikai program visszajelzett és a hibát azonnal javítani lehetett. Ez az idő csökkent azért is, mert a programírási, valamint a lyukasztási műveletek összeolvadtak. A lyukasztási művelet, mint olyan, megszűnt, lyukszalaggal, illetve lyukkártyákkal nem kellett foglalkozni az alkatrészprogram elkészítése alatt. Ezzel még a lyukasztás által elkövetett hibák is kiestek. A rendszerek a különböző manipulációs lehetőségek beépítésével nagy segítséget nyújtottak a már megírt alkatrészprogramok módosítására is. Ezekkel a manipulációs lehetőségekkel nemcsak az alkatrészprogramírás közben lehetett a programot megváltoztatni, hanem a régebben megírt alkatrészprogramokat át lehetett alakítani.

Az eddig megvalósított interaktív alkatrészprogramíró rendszerek a kitűzött célt előbbre vitték. Ennek ellenére nagyon sok lényeges problémát nem oldottak meg. Ezek a következők:

1. Nem oldották meg azt a problémát, hogy az alkatrészprogram írásához sokoldalúan képzett szakember szükséges. Alfa-numerikus display-t vagy konzolirógépet alkalmazó rendszerekben az operátor az alkatrészprogram mondatait egyenként írja, a rendszerhez tartozó tasztaturán. Az egész alkatrészprogram felépítéséről, illetve megírásáról a programozónak kell gondoskodnia. Igaz, hogy ebben segítséget kap azzal, hogy hibás utasítás megadása esetén azonnal korrigálhat, de nem mentesít a nyelv tökéletes, legapróbb részletekbe menő ismeretétől. A nyelv hiányos ismerete nagyon megnövelheti az alkatrészprogram írásának idejét, amelynek megfelelő költségkihatásai is lehetnek.
2. A rendszerek nem valósítják meg azt az elvet, amelyet az interaktív rendszernek meg kellene oldani, hogy az operátort mentesítsék a fárasztó, sok figyelmet igénylő gépies munkától. Azzal, hogy az operátornak kész utasításokat kell felépítenie, a problémát még nem oldották meg. Nagyon sok figyelmet igényel még egy alkatrészprogram utasítás teljesen hibátlan leírása is, bár ez tulajdonképpen rutinmunka. Az utasítások formailag nagyon kötöttek, a szükséges paramétereken kívül szótári szavakat, vesszőket, egyenlőségi jeleket stb. tartalmaznak, azaz kötött szintakszissal rendelkeznek. Az operátornak a feladathoz tartozó paraméterek figyelésén kívül olyan dolgokkal is foglalkoznia kell, amelyeket a számítógép is el tudna látni.

3. Nagyszámítógép igénybevétele az adott feladatok megoldására nem gazdaságos. Ugyanezt, sőt szélesebb körű feladatokat kishszámítógépen is meg lehet oldani. Ilyen megoldásról számolunk be a jelen értekezésben.

A fent összefoglalt problémák a mini-számítógépes interaktív alkatrészprogramíró rendszereknél is fennállnak. A megvalósított rendszerek többségénél az operátor hasonló manipulációkkal végzi munkáját, mint az alfanumerikus display-t alkalmazó nagyszámítógépes rendszereknél.

Interaktív rendszerek kishszámítógépeken gazdaságosabban használhatók, mint a nagyszámítógépeken. A kishszámítógépes programok megírása viszont általában költségesebb, mint a nagy - vagy közepes számítógépes programoké. Ennek az az oka, hogy a kishszámítógépre készített rendszereket fárasztó, assembly szintű nyelven kell megírni, szemben a magasabb szintű programozási eljárásokkal (FORTRAN), amelyek csak nagy - vagy közepes számítógépen alkalmazhatók gazdaságosan. Ezen csak úgy lehet igazán segíteni, ha olyan általános szintű rendszereket hozunk létre, amelyek automatizálják az egyes feladatokat megoldó rendszerek létrehozását. A jelen értekezésben ismertetett AIR program egy ilyen általános szintű rendszert képvisel. Az AIR rendszer alkalmas arra, hogy az APT család bármely tagjára interaktív alkatrészprogramíró rendszert valósítson meg.

A grafikus display-t alkalmazó interaktív alkatrészprogramíró rendszerek egészen más kategóriába tartoznak. Említettük, hogy ilyen rendszert nem volna gaz-

daságos az alkatrészprogramok kedvéért működtetni, csak kiépített CAD rendszerrel együtt. A látszólag költséges eljárás ellenére ezek valósítják meg legjobban mindazt, amit a fejtegetés során kritika tárgyává tett programok nem tudtak teljes egészében teljesíteni.

A következőkben felvázolt interaktív program használata során az operátor menük vagy funkcionális tasztaturák segítségével választja ki a geometriai alakzatot vagy valamilyen mozgásmeghatározáshoz szükséges módot. Az interaktív program által feltett kérdésekre válaszolva adja meg a szükséges paramétereket. Így a számítógép átveszi mindazt a rutin jellegű munkát, amelyet más rendszerekben eddig a programozó végzett.

9. INTERAKTIV RENDSZEREK TERVEZÉSÉNEK ÁLTALÁNOS SZEMPONTJAI

Az interaktív rendszerek a feladataikat két irányú dialógusokban oldják meg. A dialógusoknak egyik alaptípusa az irányító dialógus, ahol a cél valamilyen feltétel vagy feladat meghatározása, kikeresése. Három részből áll: Kérdés - válasz - analízis. A dialógus másik típusa az összeállító dialógus. Lényege az az akció, amelyért a rendszer létrejött. Négy részből áll: Kérdés - válasz - analízis - akció [145, 2]. A kérdéseket a számítógép, illetve a rendszer teszi fel az operátornak, a választ az operátor adja meg, míg a kérdésre adott válasz kiértékelése a számítógép feladata. Akció esetén annak végrehajtásáról szintén a számítógépnek kell gondoskodnia.

A számítógépes dialógus hardware eszközei többféleképpen lehetnek. Legegyszerűbb a számítógéphez tartozó konzolirógép. A program a kérdéseit a konzolirógépen írja ki és az operátor azon adja meg. Fejlettebb eszköz az alfanumerikus display a hozzá tartozó funkcionális, valamint alfanumerikus tasztaturákkal. Jelenleg legtöbbet nyújtja a grafikus display alfanumerikus és funkcionális tasztaturákkal, fénytollal vagy pozicionáló gömbbel.

E fejezetben azokat az alapkövetelményeket foglaljuk össze, amelyeket egy interaktív rendszertől elvárunk, illetve amelyeket szem előtt kell tartanunk [48, 11, 155, 10]. Összefoglalót adunk a rendszer típusairól és végül néhány ergonómiai szempontra mutatunk rá.

Először megvizsgáljuk az interaktív rendszerek fő tervezési szempontjait.

1. A fejezet első részében említettük, hogy egy adott interaktív rendszer célja segítségnyújtás valamilyen feladat megoldásához. A feladat megoldása dialógus közben történik, amely a számítógép és az operátor között folyik. Ezért a rendszert úgy kell felépíteni, hogy az operátor a maximális segítséget meg is kapja. Egy adott feladat bonyolultságától függően az operátort a megoldáshoz vezető legrövidebb útra kell irányítani úgy, hogy a megoldás menetét követni tudja.
2. Az összetettebb, de az egyszerűbb feladatoknak is több megoldása lehet. Az egyes megoldásokhoz különböző paraméterek és adatok szükségesek. Egy jó interaktív rendszernek biztosítani kell azt, hogy az operátor a saját intuitív döntéseivel választhassa ki azt a megoldást, amely az adott feladathoz a legalkalmasabb, vagy dönthessen olyan kérdésekben, amelyekre a rendszer önmagában nem képes. Biztosítani kell továbbá azt, hogy a rendszer felhasználója a szükséges bemenő paramétereket, illetve adatokat egyszerű módon meg tudja adni.
3. Említettük azt az alapelvet, hogy az operátor számára fárasztó és rutin jellegű munkát a számítógép végezze el. Ezzel nemcsak az operátort mentesítjük az idegölő, fárasztó munkáktól, hanem csökkentjük az elkövethető hibák számát.
4. A számítógép és az ember közötti dialógust úgy kell kialakítani, hogy a vezető szerepet a számítógép játssza. A rendszernek kell minden esetben felkínálnia az adott döntési lehetőségeket. Az adatokat a megfelelő helyen meg kell kérdeznie, hogy az operá-

tor válaszolni tudjon. Ez biztosítja az összhangot a dialógus folyamán, továbbá azt is, hogy az operátor válaszadásai minél kevesebb manipulációs munkát igényeljenek. Ezzel a feladat megoldását lerövidítjük és a dialógust rugalmasabbá tehetjük.

5. Interaktív rendszerek kialakításánál ügyelni kell arra, hogy a rugalmas kezelhetőséget biztosítsuk. Itt nemcsak az adandó válaszok egyszerű megadási módjára gondolunk, hanem arra is, hogy az operátor különböző manipulációs lehetőségekkel betekintést nyerhessen a már elvégzett munkába, részeredményeket megtekinthessen, vagy menet közben módosíthasson egyszerű eszközökkel stb.

Egy interaktív rendszer megtervezésénél, illetve megvalósításánál mindig figyelembe kell venni azt is, hogy milyen típusu operátor számára készítjük a rendszert. Más szempontokat kell figyelembe venni akkor, ha az operátor tanuló, mást, ha gyakorlott szakember. E szempontok szerint tanító (tutorial) vagy célratörő (notutorial) rendszerről beszélünk [36, 17]. A tutorial rendszer kialakításának szempontjai:

1. Az operátor nem ismeri a megoldás algoritmusát. A rendszernek éppen az a rendeltetése, hogy a megoldás módját megtanítsa.
2. A dialógus felépítésének olyannak kell lennie, hogy a logikai kapcsolatok közötti lépéseket egyértelműen feltárja. Több megoldás esetén minden megoldást meg kell ismertetni az operátorral. Ahol szükséges, magyarázó szövegekkel kell ellátni a dialógust a megértés megkönnyítése céljából.

3. A rendszer optimumát ebben az esetben a betanítás minimális ideje határozza meg és nem a megoldás legrövidebb ideje.
4. A rendszernek feladatánál és adottságainál fogva sok commentet kell tartalmaznia. Választ kell adnia az operátor kérdéseire, megmagyarázni egyes lépéseket, vagy bemutatni különböző megoldásokat, megmagyarázni az azok közti különbségeket. Így a rendszernek nagy a memóriaigénye és elég nagy futási időre is szükség van.

A notutorial rendszer fő tulajdonságai:

1. A tutoriál rendszerekkel szemben, célja a feladatmegoldás módszerét ismerő operátor számára segítséget nyújtani. A segítség kiterjed a megoldáshoz vezető ut kiválasztásának megkönnyítésére, a gépies és rutin jellegű munkák elvégzésére.
2. A rendszer optimumát a megoldás legrövidebb idő alatti elkészítése határozza meg. Ez akkor valósulhat meg, ha lehetővé teszi, hogy az operátor minél rövidebb uton megtalálja a megoldási lehetőséget.
3. Az interaktív rendszer kérdéskészletét azzal a feltételezéssel kell összeállítani, hogy az operátor a feladatmegoldások módszerét ismeri. A feladatmegoldás szempontjából közepes (átlagos) műveltségű szakembert kell feltételezni. Ezért a kérdéseket az érthetőség határán belül, mind számban, mind terjedelemben minimálisra lehet csökkenteni, hogy a rendszer dinamikája maximálisan kihasználható legyen.

A tutoriál és a notutorial rendszerek kombinált megoldását is meg szokták valósítani. A kombinált rendszerben az operátor választhatja ki egy kapcsoló vagy funkcionális gomb

segítségével, hogy a tutorial vagy a notutorial rendszerrel akar-e dolgozni. Sokszor lehetőséget adnak arra is, hogy az operátor menet közben térhessen át egyik üzemmódról a másik üzemmódra.

Egy interaktív rendszer használhatóságát a fenti elveken kívül különböző ember-gép ergonómiai tényezők is befolyásolják. Bizonyos ergonómiai tényezők a rendszer hatásfokát csökkenthetik, de ugyanakkor növelhetik is. Ilyen befolyásoló tényezők pl. a kérdésfeltevés módjai vagy formái. A válasz megadásának formái, illetve módjai, a rendszer dinamikája, amely az ember fáradékonyságára kihat. A túl lassu rendszer éppen olyan fárasztó, mint a túl dinamikus. Ezek az ergonómiai megfontolások különböző dialógus rendszereknél másként és másként jelentkeznek. Bizonyos fókig más ergonómiai szabályokat kell betartani konzolirógép, alfanumerikus vagy grafikus display perifériák alkalmazása esetén. Természetesen a dialógus módja is különböző. Konzolirógép alkalmazásánál a párbeszéd formája lényegesen kötöttebb, mint alfanumerikus display-nél.

Nevezzünk meg néhány ergonómiai megfontolást, amelyeket egy interaktív rendszer tervezésénél célszerű betartani. A következő szempontok főleg alfanumerikus display periféria alkalmazásra vonatkoznak:

1. Nagy gonddal kell megtervezni a rendszer által feltett kérdéseket. Túl hosszan megszövegezett kérdés elolvasása és megértése sok időt vesz igénybe. Ha a képernyőn egyidejűleg több, hosszan megfogalmazott kérdés jelenik meg, a kérdések áttekinthetősége csökken. A kérdések összeállításánál azt is figyelembe kell venni, hogy milyen műveltségű operátor számára készítjük a rendszert. A műveltebb operátor feltételezése esetén természetesen a feltett kérdések is tömörebbek, egyszerűbbek.

2. Kezelhetőség, rugalmasabb dialógus, valamint könnyebb áttekinthetőség végett ajánlatos, hogy egyszerre csak egy kérdés, vagy szorosan összetartozó kérdéscsoport jelenjék meg a képernyőn. Ha nincs lehetőség, hogy minden kérdéscsoportot új képernyőre írjunk ki, törekedni kell arra, hogy a képernyőn megjelenő új információ tartalma csak egy új kérdéssel térjen el a régi információ tartalmától.
3. A képernyő információ tartalmának gyors áttekintése a képernyő zsufoltságával csökken. Ezért a képernyő zsufoltságát kerülni kell, még akkor is, ha esetleg egy részfeladathoz tartozó kérdések egy képernyőre nem férnek el.
4. A dialógus dinamikája érdekében a rendszer tervezésénél törekedni kell arra, hogy a válaszokat a lehető legegyszerűbb formában adja meg. Alfa numerikus display rendszereknél a válaszokat általában két csoportra oszthatjuk. Az egyik csoportba soroljuk az olyan válaszokat, amelyekben a rendszer vagy alfa numerikus jelekből álló információt, vagy számszerű paramétereket kér az operátortól. A másik csoport, amelyben a rendszer több lehetséges kérdés közül rámutatással kéri a helyes választ. Az első csoportban azzal egyszerűsíthetjük az eljárást, hogy előírjuk a válasz pontos helyét. A második csoport erősen függ a hardwaretől. Fényceruza esetén a rámutatás a legegyszerűbb. Ha nincs fényceruza, cursor rámutatást is alkalmazhatunk. Egyszerűsödik a válaszadás, ha a kérdéseket úgy építjük fel, hogy a válaszok egy meghatározott display képernyő sorban jelenjenek meg.
5. Alfa numerikus display esetén a rendszer használatát meggyorsíthatjuk, ha a display képernyőt következő-

tesen felosztjuk. Külön helyet kell biztosítani a képernyőn a kérdéseknek, az operátor válaszainak, a rendszer egyéb üzeneteinek (hibajelzéseknek, visszajelző lépéseknek) stb.

6. A display képernyőn végzett munka sok mindenben eltér a ceruza-papír jellegű munkától. Papír-ceruzával végzett munkánál a tervező fel tudja mérni, hogy mennyi munkát készített már el, hol tart a munkával. A display képernyőjén csak kevés információ jelenhet meg egyszerre. Ezért ergonómiailag nagyon kedvező, ha a rendszer sorszámmal vagy valamilyen más módon állandóan jelzi az operátor számára, hogy a feladatmegoldás hol tart.
7. A dialógusrendszer tervezőjének vigyázni kell arra, hogy észrevehető várakozási idő ne legyen. Az operátort türelmetlenné teszi a várakozás, (nem tudja mi történt a berendezéssel), valamint a várakozásnak fárasztó hatása van.
Irodalmi adatok alapján 0,1-1 sec. közötti várakozási idők az optimálisak. 0,1 sec. alatti várakozási idő azért nem jó, mert az operátor nem veszi észre a display képernyőn végbement változást. 1 sec. feletti várakozási idő az egész rendszert tönkretetheti.
8. A dialógus felépítésénél ügyelni kell arra, hogy az operátor a tevékenységét építő jellegűnek érezze, olyan benyomásokat szerezzen, hogy az ő munkája konstruktív jellegű, saját döntéseivel viszi előre a megoldást. Ne érezze unalmasnak a munkát, lehetőség szerint a dialógust szinessé kell alakítani, kerülni kell az ürlap kitöltéséhez hasonlító monoton tevékenységet.

10. A DOLGOZATBAN JAVASOLT UJ INTERAKTIV ALKATRÉSZPROGRAMIRÓ RENDSZER

10.1 A megalapozás szempontjai

Az előbbiekben leírtuk az interaktív rendszerek általános követelményeit. A következőkben a szerző által kidolgozott új alkatrészprogramiró interaktív rendszer megalkotásának szempontjait foglaljuk össze. Ez a rendszer részletes ismertetése előtt szükségesnek látszik, mégpedig két okból. Egyrészt azért, mert enélkül maga a téma is meglehetősen definiálatlan volna, amint az irodalom kritikájáról szóló részből kitűnik, másrészt azért, mert nem csak a rendszer újdonságát kívánjuk bemutatni, hanem az ipari használhatóságát, gyakorlatias megalapozottságát is.

Az irodalmi adatok és a gyakorlati tapasztalatok azt mutatják, hogy az ipar főleg olyan rendszereket igényel, amelyek az alkatrészprogramok írását a leg-rövidebb idő alatt oldják meg azzal, hogy maximális segítséget adnak az operátornak. Ezek az APT típusu nyelvek minimális ismeretét kívánják meg az alkatrészprogramozótól.

Röviden összefoglaljuk azokat a minimális követelményeket, amelyek ismeretével rendelkezni kell ahhoz, hogy a legjobb hatásokkal használhassa a dolgozatban javasolt és a kutatás során megvalósított interaktív alkatrészprogramiró rendszert.

- A nyelv felépítésének minimális ismerete.
- A használni kívánt nyelv utasításkészletének általános ismerete.

Az utasítások pontos formáját és szintaktikus szabályait azonban nem kell ismerni.

- A nyelvben használt azonosítók alkalmazása.
- A programozást megkönnyítő utasítások használata (MACRO, LOOP stb).

Az összefoglaltak elégségesek ahhoz, hogy az interaktív alkatrészprogramíró rendszerrel a programozó alkatrészprogramot készíthessen. A leírt követelmények viszont nem volnának elégségesek alkatrészprogramok önálló megírására az interaktív rendszer nélkül.

Ahhoz, hogy az operátor az alkatrészprogramíró rendszert használni tudja, szükséges a rendszer kezelését, valamint annak főbb lehetőségeit megtanulni. Ezek a következők:

1. Hardware berendezések kezelése. (A programok beolvasása a számítógépbe, a display és az alfanumerikus tasztatura kezelése).
2. A válaszok megadásának módjai (szöveges információk, rámutatás).
3. A dialógusokon belül történő visszalépés formája, illetve hatása (VISSZA rámutatás).
4. A rendszerbe beépített manipulációs lehetőségek használata.

A minimális ismeretek elsajátítása maximálisan 1-2 nap. A technológiában járatos szakember tehát komoly időráfordítás nélkül írhat alkatrészprogramot.

Az interaktív alkatrészprogramíró rendszer biztosítja azt, hogy az elkészült alkatrészprogram szintaktikus hibáktól teljesen mentes és a szemantikus hibák előfordulásának valószínűsége csak nagyon kicsi.

Ahhoz, hogy az alkatrészprogramíró rendszer a ráruhozott feladatokat jól meg tudja oldani, pontosan specifikálni kell azokat a követelményeket, amelyek biztosítják a sikeres munka feltételét. A specifikáció összeállításánál abból az alapeleggondolásból indultunk ki, hogy az alkatrészprogramban szereplő utasítások megírását nem szabad az operátorra bízni. Ez nemcsak a nyelv tökéletes ismeretét kívánná meg az alkatrészprogramozótól, hanem megnövelné a hibák lehetőségét és az alkatrészprogram megírásának az idejét is meghosszabbítaná. Minden utasítást a rendszernek kell felépítenie. Egy utasítás kiválasztásával, az abban szereplő fő szó, illetve módosítók nagyrésze meghatározott. Az operátorra csak az utasításban szereplő paraméterek (azonosítók) és számszerű értékek megadását, valamint a módosítók kiválasztását szabad bízni. A paraméterek megadásának több módja lehet. Ezek közül csak egy néhányat emelünk ki:

- Az utasításokban szereplő paraméterekre táblázatokat készíthetünk és az operátor ezeket a táblázatokat tölti ki.
- A paraméterekre kérdőíveket állítunk össze, amelyek kitöltésével kapja meg a rendszer a kért értékeket.
- Az utasításokat paraméterek nélkül kiíratjuk úgy, hogy a hiányzó adatoknak a helyét kihagyjuk. Az operátor a szabad helyekre írja be az aktuális paramétereket.
- A programozó a kívánt paramétereket párbeszéd keretén belül adja meg.

Az utolsó pontban leírt eljárás elégíti ki legjobban egy rugalmas interaktív rendszer igényeit. Ezért ezt a megoldást választottuk.

Az operátortól megkapott aktuális paraméterek segítségével a kívánt utasítást az alkatrészprogramíró rendszer állítja össze.

A jobb áttekinthetőség kedvéért az alkatrészprogramíró rendszer követelményét pontokban foglaljuk:

1. Minden APT típusu nyelvben szerepel néhány olyan utasítás, amelyet feltétlenül meg kell adni. Ilyen például a PARTNO, a MACHIN, vagy a megmunkálandó munkadarab anyagának jele vagy száma, maró megmunkálás esetén a maró szerszám átmérője, a szerszám indulási helyzete stb.

Az alkatrészprogramíró rendszernek biztosítani kell azt, hogy az adott APT típusu nyelvben szereplő összes olyan utasítások bekerüljenek a programba, amelyeket az adott nyelv kötelezően előír. Ezt legegyszerűbben úgy érhetjük el, hogy az egyéb utasítások összeállítását csak akkor engedjük meg, ha már a kötelező utasítások meghatározása megtörtént.

2. Biztosítani kell azt, hogy az operátor a nyelvben szereplő bármely utasítást a legegyszerűbb módon találhassa meg. Erre azért van szükség, hogy a kiválasztott rekordot minél hamarabb összeállíthassa. Egy utasítás kiválasztása az utasítások halmazából legegyszerűbben több szintű menü szervezéssel valósítható meg. Biz-

tosítani kell továbbá azt, hogy egy utasítás összeállítása után a következő utasítás megkereséséhez mindig csak az a menü szint jelenjék meg az operátor előtt, amely a kívánt utasítás kiválasztását egyértelműen a legrövidebb uton határozza meg. Ez a feltétel nem csak az alkatrészprogram megírási idejét csökkenti, hanem a munkát sokkal gördülékenyebbé is teszi. A menüből rámutatással lehet választani. A hardware lehetőségektől függ, hogy milyen módon, pl. fénytollal, pozicionáló gömbbel vagy tasztaturáról vezérelt cursorral.

3. Az APT típusú nyelvek utasításaiban szereplő paramétereket három féle válaszadási móddal lehet megadni. Az első mód a rámutatás. Ezt akkor használjuk, ha több, de nem túl nagy számú lehetőség közül választjuk ki a megfelelőt.

A második mód, amikor a válaszadás szövegszerű. E típusú válasszal adja meg az operátor az utasításokban szereplő azonosítókat vagy commenteket stb. A harmadik mód az adatok számszerű megadása.

4. Az interaktív rendszernek kell gondoskodnia arról, hogy az operátor által megadott paraméterek, egyrészt szintaktikusan hibátlanok legyenek, másrészt szemantikus hibák se kerüljenek az alkatrészprogramba.

Két módszer közös alkalmazását javasoljuk. Az egyik módszerrel a válasz megadásának módját és formáját olyanra választjuk meg, hogy az operátor hibát ne követhessen el.

A másik módszer, hogy a rendszer által beolvasott válaszokat egy diagnosztikai programrész megvizsgálja. Ez csak azokra a hibákra korlátozódik, amelyeket az első módszerrel nem tudunk kiszűrni.

Összefoglaljuk, milyen hibatípusokat tudunk kiszűrni az egyes módszerekkel.

- Módosítók meghatározásának hibái. Kérdés formában felajánljuk mindazokat a módosítókat, amelyek az adott utasításban az adott helyen alkalmazhatók. Az operátor a felkinált módosítók közül rámutatással választhatja ki a megfelelőt.
- Hiba lehet a paraméterek hosszának túllépése. Egy utasításban szereplő - (az operátor által szabadon adható) - azonosítók 6 karakternél hosszabbak nem lehetnek. Ha a válaszadás lehetőségét korlátozzuk a display képernyőre felrajzolt két jel közé, úgy hogy 6 karakternél többet ne lehessen beírni, az operátort rákényszerítjük e kötöttség betartására.
- Hibalehetőség a comment szövegek helytelen használata.
A Comment szövegek hosszának betartására a 2. pontban ismertetett módszert használhatjuk, természetesen az APT típusu nyelv ide vonatkozó kötöttségének megfelelően.

A leggyakrabban előforduló hibákat a diagnosztikai programnak kell kiszűrni. Ezek a következők:

- Az operátornak a feltett kérdésekre minden esetben válaszolnia kell, kivéve akkor, ha a

rendszer a válasz kihagyására külön engedélyt ad. Ha az operátor a feltett kérdésre nem ad választ, a rendszernek hibajelzéssel kell reagálnia.

- Meg kell vizsgálni, hogy az operátor által adott azonosító számmal kezdődik-e vagy sem. (Azonosító számmal nem kezdődhet).
- Utasítás megnevezésére használt azonosító az alkatrészprogramban nem szerepelhet.
- A definiáló utasítás azonosítójának már az alkatrészprogramban előzőleg szerepelnie kell, mielőtt más utasításban arra hivatkozás történik.

5. A képernyő előtt dolgozó operátor számára a munkát rugalmassá, illetve gördülékennyé kell tenni. Ezt a feltételt elégítjük ki akkor, hogy ha a rendszer az operátor választ minden esetben nyugtázza. Ha a nyugtázás elmarad, ez rendszerhibára utal. Az operátor a nyugtázás nyomán érzékeli azt, hogy valami történt az ő beavatkozására. A nyugtázás több formában történhet.

Ha a rendszer hibátlan választ kapott a feltett kérdésre, a reagálás kétféle lehet. Amennyiben az APT utasítás felépítése még nem fejeződött be, úgy újabb kérdés megjelenésével az operátor tudomást szerez arról, hogy az adott választ a rendszer elfogadta. Ha az utasítás felépítése befejeződött, a kész összeállított utasítást kell kiírni. Ennek jelentősége kettős. Egyrészt a fenti követelménynek eleget teszünk, azaz az operátor értesül arról, hogy hibátlan választ adott, másrészt az összeállított utasítást egy végső ellenőrzésre ajánljuk fel.

Ha a rendszer hibás választ kapott, minden esetben hibajelzést kell kiírni. Ezzel eleget teszünk annak a feltételnek, hogy a rendszer nyugtázza a hibás választ és egyben megadhatjuk a hiba kijavításának a módját is.

6. A tárgyalt hibatípusokon kívül a válaszokban előfordulhatnak olyan hibák, amelyeket a rendszer sem ki-
védeni, sem ellenőrizni nem képes. A rendszernek meg kell teremtenie azt a lehetőséget, hogy az operátor ezeket a hibákat kijavíthassa még a munkája közben, ha ezeket észreveszi. Arra kell törekedni, hogy a hiba kijavítására olyan módszereket ajánljunk fel, amelyekkel gyorsabban javíthatók. A felajánlott módszerek természetesen annak a függvényei, hogy a hibát az alkatrészprogram írásnak mely stádiumában észlelik. Ha az operátor téves információt ír le és a hibát észreveszi, mielőtt azt a számítógépbe beküldené, a hardware megadja a lehetőséget az azonnali javításra. Ha a hiba észlelése egy befejezetlen, ill. egy el nem fogadott utasítás előtt történik, azaz egy cikluson belül, meg kell adni azt a lehetőséget, hogy az javítható legyen. Erre javasoljuk azt a megoldást, hogy a képernyő egy bizonyos helyére történő rámutatással az operátor visszaugorhasson az elkezdett ciklus elejére és azt újra kezdve még egyszer felépíthesse úgy, hogy a cikluson belül addig megadott információk törlődjenek. Ha a hibát nem az adott ciklus felépítése közben veszik észre, annak javítása csak a rendszerbe beépített úgynevezett manipulációs lehetőségekkel lehetséges. (Lásd 8. pont).

7. Az alkatrészprogramírás befejezésekor a rendszernek egy néhány feladatot kell elvégeznie.

Ezeket összefoglalva:

- Teljes alkatrészprogram ellenőrzése. A fent említett hibákon kívül a következő ellenőrzések végrehajtására lehet felkészíteni a rendszert:
 - a. Az olyan definiáló utasítások kiszűrése, amelyekhez nem tartozik más utasításban hivatkozás.
 - b. Olyan zárt utasításegységek ellenőrzése, amelyek nyitáson kívül záró utasítást is igényelnek (pl. MATRIX - TERMAC, CONTUR - TERCON). Az ellenőrzés tárgya az, hogy van-e záró utasítás.
 - c. Ellenőrzendő, hogy a MACHIN utasításban megnevezett posztprocesszor szerepel-e a posztprocesszor listán.
 - d. Vizsgálni kell, hogy a posztprocesszor által eltiltott vagy korlátozott utasítások szerepelnek-e az alkatrészprogramban. Ha szerepelnek, hibajelzést kell kiírni.
- A hibátlan alkatrészprogramról listát kell ki-nyomtatni.
- Az alkatrészprogramról lyukszalagot vagy mágnesszalagot kell készíteni olyan kódrendszerben, amelyet a feldolgozó számítógép igényel.

A leírt ellenőrzések beépítése nem kötelező, de annál kiépítettebb a rendszer, minél többet bízunk rá.

8. Interaktív rendszer használhatóságát (hibajavítás) és rugalmasságát nagyon megnövelhetjük, ha különböző manipulációs lehetőségeket építünk be. Néhány ma-

nipulációs műveletet sorolunk fel. Itt alfanumerikus display alkalmazását tételezzük fel.

- Az operátor által már megírt utasítások közül, akár sorszámmal, akár szimbólummal azonosított utasítás iratható ki a display képernyőre.
- A már megírt utasítások közül sequenciálisan utasításokat lehet megjelentetni egyszerre annyit, ahány egy képernyőre kifér.
- Tetszőleges két utasítás közé egy vagy több utasítást lehet beszurni.
- A már megírt alkatrészprogramból tetszőleges utasítást lehet törölni.
- Az alkatrészprogramban használt azonosítókról táblázatot lehet kiíratni a display képernyőre.
- Az alkatrészprogramban szereplő jellegzetes utasítások kiirathatók (FROM, MATRIX stb.).
- Néhány APT típusu rendszer alkatrészprogramjába olyan kódszámok beírását engedi meg, amelyekre táblázatok állnak rendelkezésre. Pl. az EXAPT1-ben a menetfajta kódtáblázat, vagy az orsó visszafutás kódtáblázat. Az ilyen típusu táblázatok kiíratása a display képernyőre.
- Minden rendszer alkalmazása bizonyos foku kötöttséget jelent a felhasználó számára. Ezek a kötöttségek sok esetben nehézkessé teszik az alkalmazást. A legjobban átgondolt rendszert sem lehet mentesíteni az ilyen jellegű korlátoktól. Ha kiépítünk egy olyan lehetőséget, amely lehetővé teszi a kilépést az interaktív rendszerből úgy, hogy az operátor APT típusu utasításokat írhasson dialógus felhasználása nélkül, a rendszer rugalmasságát megnöveljük a gyakorlott operátor számára.

10.2 A rendszer felépítése, szervezésének strukturája

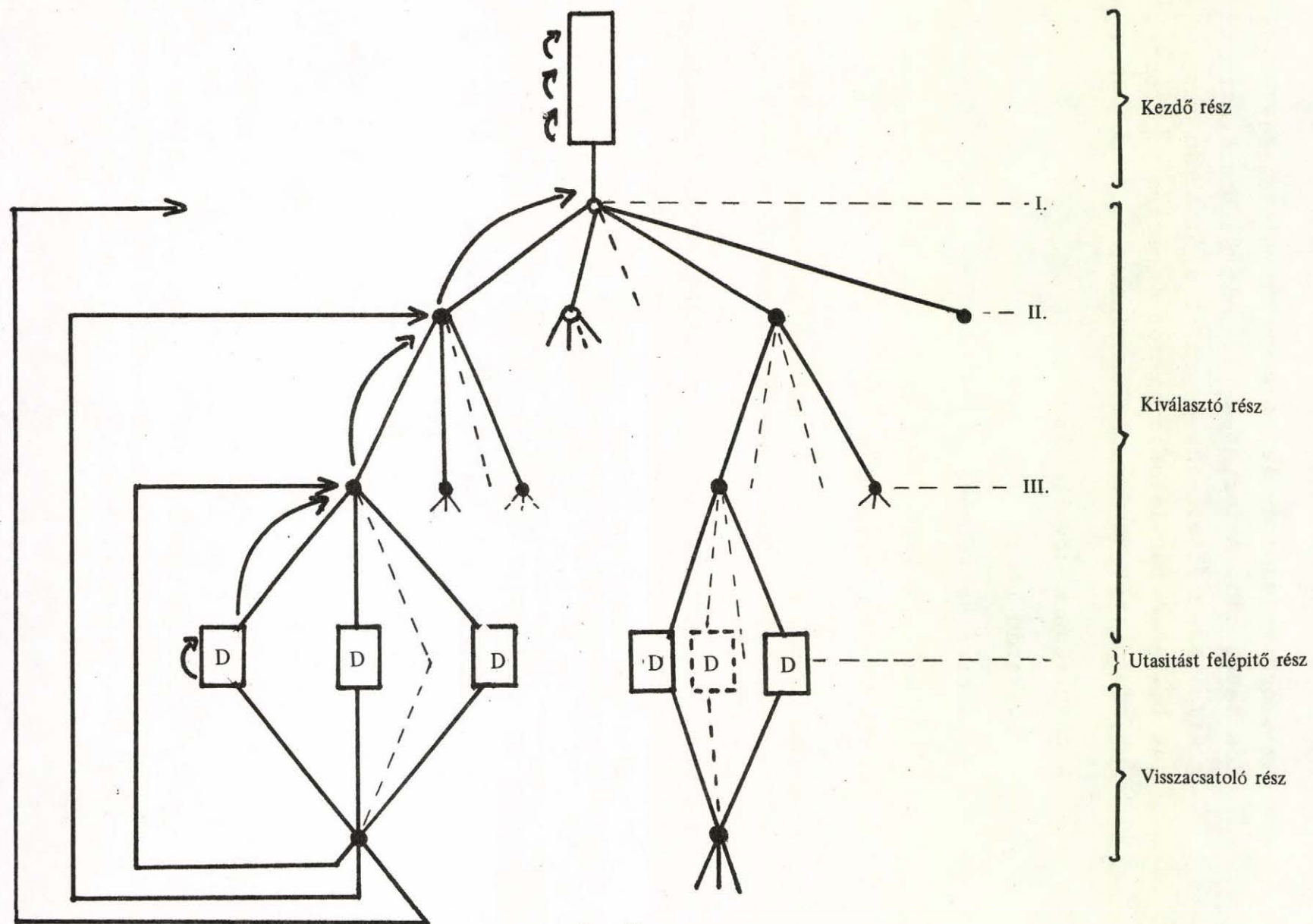
Az előző fejezetben leírt sokrétű követelmény megvalósítása, ill. áttekintése egy eléggé összetett gráffal szemléltethető. A gráfot a 6. ábrán mutatjuk be. Az ábrán látható gráfot 4 részre tagolhatjuk, felépítés és funkció szerint.

A négy rész a következő:

1. kezdő rész
2. kiválasztó rész
3. utasítást felépítő rész
4. visszacsatoló rész.

A kezdő részben utasításokat összeállító dialógusok sequenciálisan helyezkednek el. Ide az olyan utasítások dialógusát helyezzük el, amelyeket minden alkatrészprogramban kötelező megadni. A sequenciálisan felépített dialógusokban menük és adatokra vonatkozó kérdések felváltva szerepelnek. A dialógusokon belül kisebb visszacsatolások szerepelnek a hibajavítások könnyebb elvégzése miatt.

A kiválasztó rész felépítését megközelítőleg egy háromszintű fa struktúra jellemzi. Maximálisan háromszintű fa struktúra elegendő ahhoz, hogy az APT típusu nyelvek utasításait olyan megfelelő csoportosításba soroljuk, hogy azokat könnyen meg lehessen találni, illetve kiválasztani. Nem szükséges minden utasítás kiválasztáshoz a fa struktúra mindhárom szintje. A fa struktúra csomópontjai alkotják az elágazásokhoz tartozó menüket, amelyek segítségével a megfelelő utasítások kiválaszthatók. A kiválasztó rész nem nevezhető teljes mértékben fa struktú-



6. ábra

rának, mivel visszacsatolások vannak benne az alacsonyabb szintről a magasabb szintre. A visszacsatolásokra azért van szükség, hogy téves kiválasztás esetén a javításra lehetőséget biztosítsunk.

Az utasítást felépítő része fa struktúra ágainak végén helyezkedik el. E részben található az utasítások felépítéséhez tartozó összeállító dialógusok. A dialógusok sequenciálisan követik egymást, menük és adatokra való kérdés változtatásával az adott utasítás felépítésétől függően. E részben szintén szükséges a hibajavítás lehetőségének biztosítására kisebb visszacsatolásokat beépíteni.

A gráf utolsó részén helyezkedik el az ugynevezett visszacsatoló rész. A visszacsatoló rész felfogható egy egyszintű fa struktúrának, amelynek ágai a kiválasztó rész egyes szintjeire mutatnak. Az alkatrész-programírás ciklikussága teszi szükségessé e rész kialakítását. Nem engedhető meg az, hogy minden utasítás kiválasztása a kiválasztó rész első szintjéről történjék abban az esetben is, ha arra nincs szükség.

10.3 APT típusu nyelvek utasításainak felosztása a gráfon belül

Az APT típusu nyelveket ismertető manuelek bizonyos szempontok szerint csoportosítják a hozzájuk tartozó utasításokat. E csoportosítások már a rendszerek különbözőségei miatt sem egységesek és a csoportosítás szempontjai sem felelnek meg az interaktív programozás követelményeinek. Az egyes rendszerek csoportosítását főleg a processzoraikban történt feldolgozás sorrendje, valamint a nyelv könnyű megtanu-

lásának szempontjai határozták meg.

Az interaktív alkatrészprogramíró rendszernél az utasítások felosztását az alkatrészprogram írásának szempontjai határozzák meg. Röviden összefoglaljuk ezeket a szempontokat:

1. Feltételezzük, hogy az alkatrészprogramíró a gyakorlatban a legjobban bevált és az alkatrészprogramot legjobban áttekinthető módszer szerint írja. E szerint először leírja azokat az utasításokat, amelyek megadása az alkatrészprogramban kötelező. Ezután határozza meg a megmunkáláshoz szükséges geometriát, majd a technológiát. Az egyes technológiai definíciók vagy egyéb funkciók aktivizálása után írja le a szerszám mozgás utasításait. E sorrend betartása azt jelenti, hogy egy csoportba tartozó utasításokat úgy kell összeállítani, hogy lehetőség szerint az egymás után írt utasítások egy csoportba legyenek. Ez az alkatrészprogramírás idejét nagyon lerövidítheti.
2. Funkciókban hasonló típusu utasításoknak a könnyebb kiválasztás miatt azonos csoportba kell kerülniük.
3. Az olyan utasításokat is célszerű egy csoportba tenni, amelyek funkciójukban ugyan eltérnek egy adott csoport utasításaitól, de alkalmazásuk a többi utasítással együtt gyakori.
4. Bármely utasításra biztosítani kell a könnyű hozzáférhetőséget.

A fenti szempontok figyelembevételével irányelveket állítunk össze arra, hogy az APT típusu nyelvekben

szereplő utasításokat milyen felosztásban célszerű elhelyezni és arra, hogy ezek a csoportok a gráfban hol helyezkednek el.

A kezdő részben helyezzük el az említettek szerint azokat az utasításokat, amelyek megadása feltétlenül szükséges ahhoz, hogy a processzoron az alkatrész-program futtatható legyen. Az utasítások közül csak egy néhányat emelünk ki, például: PARTNO, MACHIN vagy NOPOST, TRANS, FROM stb. Ezek minden APT típusu rendszerben szerepelnek.

Mivel az APT család tagjai nem egyformák, vannak olyan utasítások, amelyek az egyik nyelvben kötelezően szerepelnek, a másik nyelvben pedig nem. Például: a PART utasítás az EXAPT1-ben kötelező, míg a 2CL-ben nem, viszont a 2CL-ben kötelező a CUTTER utasítás, míg az EXAPT1-ben nem is szerepel.

A kiválasztó rész fa strukturájának első szintjéből kiinduló ágak határozzák meg az utasítások felosztásának fő csoportjait. Ezek a következők:

1. Geometriai utasítások csoportja.
2. Technológiai utasítások csoportja.
3. Egyéb programozástechnikai utasítások.
4. Lehívó, illetve végrehajtó utasítások csoportja.
5. Mozgásutasítások csoportja.
6. Interaktív manipulációs lehetőségeket kiválasztó csoport.
7. Alkatrészprogram vége.

A felsorolt pontokból összeállított lista egyben az utasítások kiválasztásának első menüjét adja meg.

Nézzük meg milyen utasításokat, illetve funkciókat helyezünk el az egyes csoportokba.

1. Geometria. Ebbe a csoportba helyeztünk minden olyan utasítást, illetve utasításcsoporthoz, amely a geometria leírását hivatott definiálni. Ezek az utasításcsoporthoz a következők lehetnek, attól függően, hogy az interaktív alkatrészprogramíró rendszert mely APT típusú rendszerre hozzuk létre.

1. Sik meghatározásának utasításai
2. Pont utasításai
3. Pontmintázat utasításai
4. Egyenes utasításai
5. Kör utasításai
6. Kupszeletek utasításai
7. Táblázatosan adott görbék utasításai
8. Vektorok meghatározásának utasításai
9. Koordináta-rendszer transzformációi.

A létrehozott utasításcsoporthoz több utasítás tartozik, attól függően, hogy az adott nyelv egy geometriai elem meghatározására hány lehetőséget ad (pl. 2CL az egyenes meghatározására 10 lehetőséget enged meg, szemben az EXAPT1-el, ahol csak 3 lehetőség van.)

A koordináta-rendszer transzformációt definiáló utasítás szorosan nem tartozna ebbe a csoportba, de mivel a geometriai utasításokkal együtt használják, célszerű ebbe a csoportba elhelyezni. A fa struktúra harmadik szintje

vagy a harmadik menü az egyes geometriai elemek elágazásánál jön létre (lásd a 6. ábrát). A harmadik szinten az elágazások számát mindig az adott csoportba tartozó utasítások száma határozza meg. Ebből a szintből kiinduló ágak közvetlenül az operátor által kiválasztott utasításra mutatnak rá.

2. Technológia utasítások csoportja. Ide azokat az utasításokat vagy utasításcsoportokat célszerű elhelyezni, amelyekkel a megmunkáláshoz szükséges technológiai adatokat írjuk elő. Az APT család tagjai e területen nem mutatnak olyan egységes képet, mint a geometriai fejezet utasításai. Vannak olyan APT típusu rendszerek, amelyekben a technológia automatizálása vagy annak nagy része megoldott, (pl. EXAPT rendszerek vagy a NELNC rendszer), de vannak olyanok, amelyekben a technológia nincs automatizálva, így csak a posztprocesszor mondatok között van lehetőség néhány egyszerű technológiai adat megadására, (pl. IFAPT, APT). A fa struktúra második szintjének elágazását természetesen az adott rendszer technológia automatizálása határozza meg. Ebből a szempontból fejlett rendszerre a következőt javasoljuk, (EXAPT rendszerekre):

1. Ciklus megmunkálások előírása
2. Egyedi megmunkálások előírása
3. Járulékos technológiai adatok megadása.

Ciklus megmunkáláshoz azokat az utasításcsoportokat soroljuk, amelyekkel olyan technológiai előírásokat lehet leírni, amelyekhez több egyedi megmunkálás tartozik (pl. menetfurás).

Az egyedi megmunkálást leíró utasításokkal csak egy technológiai művelethez tartozó paramétert lehet programozni, például előtolás, fordulatszám stb.

Mindhárom megmunkálási típus csoporthoz külön-külön több utasítás tartozik. E csomópontokhoz tartozó elágazások alkotják a fa struktúra harmadik szintjét, amelyekből a kiinduló ágak közvetlenül a kívánt utasításokra mutatnak.

3. Egyéb programozástechnikai utasítások. Az APT család minden tagjához tartozó nyelvben szerepelnek olyan utasítások, amelyeknek az a céljuk, hogy a programozó munkáját megkönnyítsék. Természetesen egységességről itt sem lehet beszélni. Vannak olyan rendszerek, ahol az ilyen jellegű utasítások száma nagyobb, de vannak olyanok, ahol ezek száma kevesebb. Teljes listát ezekről az utasításokról nagy számuk miatt nem adunk, csak a legfontosabbakat említjük:

1. Megjegyzések az alkatrészprogramban
2. Utasítás kiíratása a gépkezelő számára
3. Zárt konturok definiálása
4. Hurok utasítások
5. MACRO, illetve szubrutin utasítások az alkatrészprogramon belül
6. Aritmetikai utasítások.

Ezeknek az utasításoknak a száma lényegesen kevesebb, mint a geometriai utasításoké. Így nem feltétlenül célszerű a fa struktúra harmadik szintjét kialakítani. A második szint már közvetlenül a kívánt utasításra mutathat rá.

4. Lehívó, végrehajtó utasítások csoportja.

E csoportba tartozó utasítások szintén az adott APT típusu rendszerbe beépített utasítások függvényei. Jellemükénél fogva többfélék. Így például a programozástechnikai utasításokban definiált rekordok (pl. MACRO utasítás lehívása), technológiai utasításokban meghatározott funkciók vagy a szerszámgép valamely funkciójának aktivizálása (pl. hűtés be-, kikapcsolása, szerszámgép leállítása stb). A felsorolt példákból is látható, hogy nem minden esetben szükséges a fa struktúra harmadik szintjét kialakítani, csak akkor, ha az adott rendszernél célszerű. Ezt a szerszámgép nagyszámu egyéb funkciói megkövetelhetik.

5. Mozgásutasítások. Ide soroljuk a szerszámmoknak, vagy a szerszámgépek asztalainak mozgatási utasításait és célszerűen a felületek vagy konturok kialakításának minden egyéb mozgásutasítását. Ezek száma APT családonként változó. Például egy háromdimenziós megmunkálás leírására kialakított rendszer sokkal több és más jellegű mozgásutasítást tartalmaz, mint egy szakasz vezérlésű megmunkálás. Ezen belüli újabb csoportok kialakítása természetesen a rendszer bonyolultságától függ. Kevésbé bonyolult rendszernél a fa struktúra harmadik szintje elhagyható.

6. Manipulációs lehetőségek. Az e csoporthoz tartozó faágak az interaktív rendszer használatát megkönnyítő manipulációs lehetőségek felé mutatnak. Az innen kiinduló manipulációs lehetőségeket az "Interaktív alkatrészprogramíró rendszer követelményei" című fejezetben már összefoglaltuk.

A fent leírt csoportok kialakításával elérjük azt, hogy az operátor a kívánt utasítást a fa struktúra két vagy három szintjének végigjárásával megtalálja. Az ágak végén kell elhelyezni azt az egy vagy több tagból álló dialógus láncot, amellyel a programozó az utasítást fel tudja építeni.

Az EXAPT1 nyelv utasításainak felosztását az 1. sz. függelék tartalmazza a manipulációs lehetőségek felsorolásával együtt.

10.4 A dialógusok megtervezésének szempontjai

Interaktív alkatrészprogramíró rendszer általános szempontjait az előzőekben már ismertettük. Ebben a fejezetben azokkal a módszerekkel foglalkozunk, amelyekkel meg tudjuk valósítani az ott leírtakat.

Az ember-számítógép alfanumerikus display-n folyó párbeszédhez, az alkatrészprogramíró rendszerben az alábbi fő információ típusok tartoznak:

1. Tájékoztató jellegű információk.

Nagyon sokszor a display képernyőn megjelenő kérdésekből nem lehet egyértelműen meghatározni, hogy milyen típusú utasítást állít össze az operátor, azaz a kérdések milyen típusú utasításhoz tartoznak. Célszerű a képernyőn feltüntetni annak az utasításnak a fő szavát, amelynek írása éppen folyik.

Ergonómiai összefoglalónkban beszéltünk arról, hogy az operátor display képernyő előtt ülve, nem tudja áttekinteni, hogy mennyi munkát végzett már el. Az utasítások sorszáma függvénye ennek, így célszerű ezt is kiírni a képernyőre.

A leírt két tájékoztató jellegű információon kívül lehetnek még egyéb információk is, de ezt a kettőt tartjuk a leglényegesebbnek. Ezeket a képernyőn célszerű mindig ugyanarra a helyre kiírni a jobb áttekinthetőség miatt.

A megvalósított rendszerünkben a képernyő első sorának jobb, illetve bal végét használtuk fel.

2. Munkaterület.

A display képernyőn munkaterületet kell kialakítani a rendszer és az alkatrészprogramozó közti dialógus céljára. A munkaterület alakja az adott feladathoz alkalmazkodik. Az interaktív párbeszéd során a rendszer kérdéseket tesz fel, amelyre az operátornak válaszolnia kell. Mind a kérdés, mind a válasz módja többféle lehet. A kérdés vagy egy paraméterre irányul, vagy a rendszer menüt ajánl fel az operátornak. A válasz ennek megfelelően vagy paramétermegadás, vagy rámutatás a felajánlott menü egyik tagjára.

A megoldás jellemzői:

- A rendszer által feltett kérdéseket a display képernyő egy bizonyos területén helyezzük el, célszerűen a munkaterület baloldalán.
- Az operátor által adott válaszok helyét két jellel (kezdő és végjel) ki kell jelölni, szintén a munkaterület egy állandó részén. A megvalósításban a képernyő jobboldalán ugyanabban a sorban, amelyben a kérdés megjelent.
- A képernyőn egyszerre csak egy új kérdés jelenik meg a hozzátartozó válasz hely kijelölésével. Menü esetén a menü elemei lehetőleg külön-külön sorban jelennek meg úgy, hogy a rámutatás mezői is elférjenek mellettük.
- Általában egy APT típusú utasítás felépítése több dialógusból áll. Egy utasításhoz tartozó kérdések egymás után jelennek

meg. Az előző kérdéseket a következő megjelenésekor nem törli a program, hogy az operátor könnyen áttekinthesse a szóban forgó utasítás addig megadott paramétereit.

A munkaterület céljára felhasználható a képernyő 2. sorától az egész display területe az utolsó két sort kivéve.

3. Visszacsatolás a dialóguson belül.

Lehetőséget kell adni az operátornak arra, hogy esetleges téves válaszait korrigálhassa és a dialógusban visszaléphessen. Erre a megvalósított rendszerben a képernyő utolsó előtti sorára történő rámutatást alkalmazzuk egy "VISSZA <>" felirattal. Ha erre a sorra mutat rá az operátor, egy logikai szinttel ugrik vissza a dialógusprogram.

4. A rendszer üzenetei.

A képernyő utolsó sorában helyezzük el az operátornak szánt üzeneteket, például hibajelzések magyarázattal, kész utasítások kiírása, vagy egyéb üzenetek.

Összefoglalva a 7. ábra mutatja a képernyő megvalósított felosztását.

A display képernyő területi felosztásán kívül még néhány szempontra szeretnénk felhívni a figyelmet, amelyekkel az operátor munkáját megkönnyíthetjük. Menü típusu kérdés esetén a rámutatás módjának kiválasztása bizonyos fokig a hardware adta lehetősé-

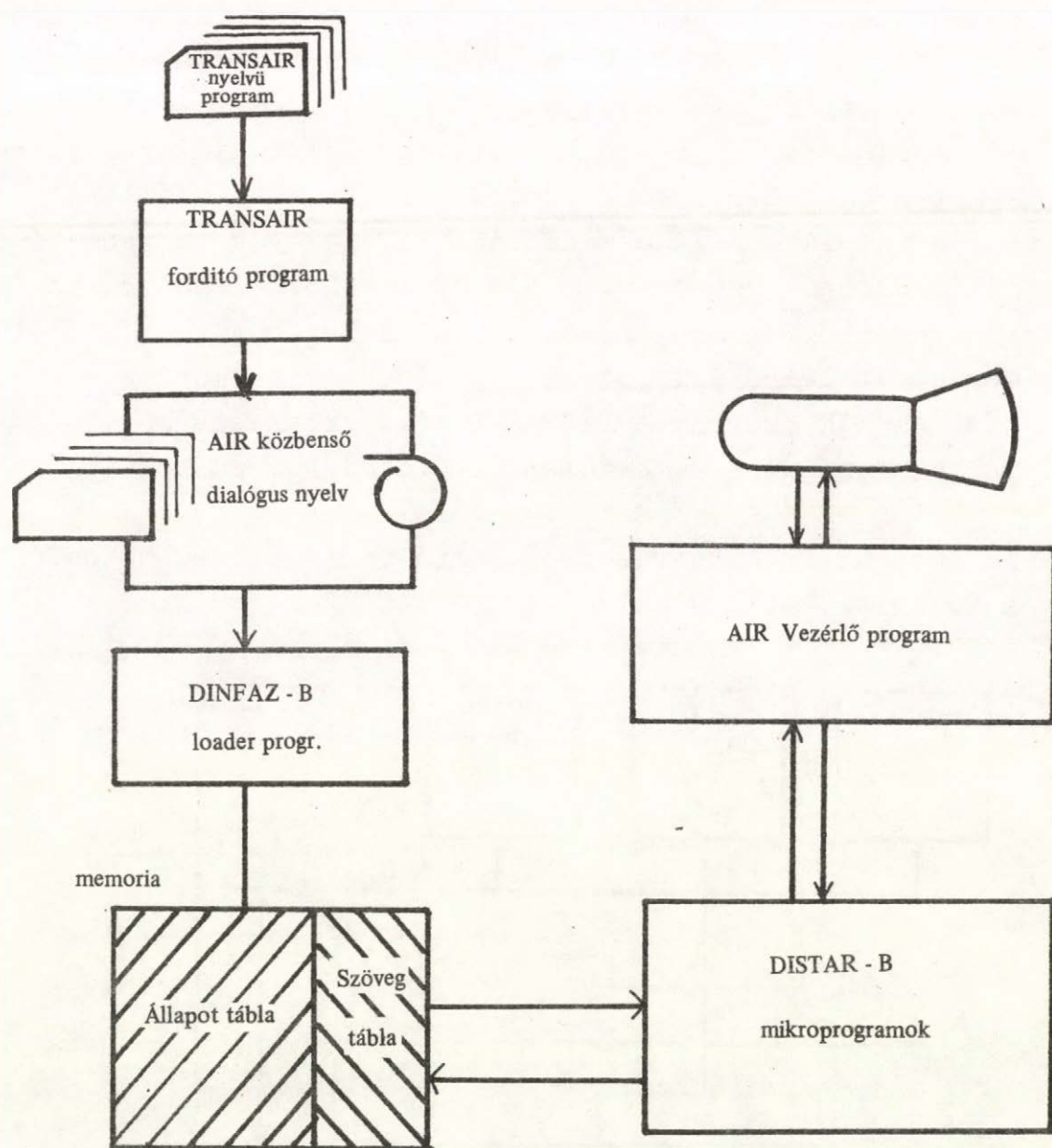
Tájékoztató jellegű információk	1
Munkaterület	2
	3
	14
Kérdések	
Visszacsatolás a dialóguson belül. VISSZA Δ Δ	15
A rendszer üzenetei.	16

7. ábra

gektől függ. Az irodalomban nagyon sok megoldást találunk erre. (Fényceruzával, pozicionáló gömbbel, cursor rámutatás, menü elemet jelző sorszám adott helyre beírása stb.). Rendszerünkben a cursor rámutatást választottuk. A display képernyő megfelelő karakter helyére visszük a cursort és az SOH tasztatura gomb benyomásával a cursor cím (sor és oszlop sorszám) bekerül a számítógépbe. A rámutatás megkönnyítése és meggyorsítása érdekében a rendszert úgy alakítottuk ki, hogy a cursort nem szükséges a kijelölt karakterhelyre vinni, elegendő csak a kérdést tartalmazó adott sorra tenni. A válaszadást meggyorsíthatjuk azáltal is, hogy a cursort a rendszer vigye arra a helyre, ahová a választ várja, illetve menü esetén a legvalószínűbb válaszadás helyre. Ezzel azt az időt takarítjuk meg míg az operátor a kijelölt helyre viszi a cursort. Az EXAPT1 interaktív alkatrészprogramíró rendszer néhány dialógusképét mutatjuk be a 2. sz. függelékben.

11. AZ AIR ALKATRÉSZPROGRAMIRÓ RENDSZER

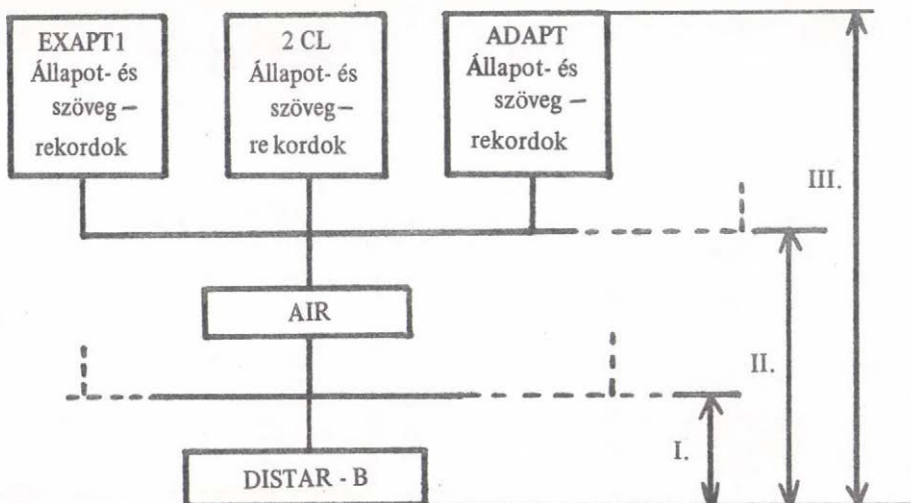
Az interaktív alkatrészprogramiró rendszer felépítését a 8. ábra mutatja.



8. ábra

Az alkatrészprogramíró rendszer két részből tevődik össze. Az első részben a dialógusokhoz szükséges adatok összeállítása, a második részben a dialógusok végrehajtása történik. A 8. ábra alapján a dialógusokat kétféle módon lehet leírni. Az első esetben a magasabb szintű TRANSAIR dialógus nyelven megírt dialógus programot a TRANSAIR fordító program olvassa és dolgozza fel. A TRANSAIR fordítóprogram kimenete a direkt kódú AIR közbelső nyelvű dialógus program, amely összeállítható közvetlenül is. Közvetlenül a direkt kódú AIR közbelső nyelven megírt dialógus program a dialógus leírásának másik módja. Az AIR közbelső nyelven megírt dialógus programot a DINFAZ-B loader program olvassa be és állítja elő a memóriában (lemezmemórián) a dialógus végrehajtásához szükséges adatokat. A dialógus végrehajtásáról az AIR vezérlőprogram gondoskodik a memóriában tárolt, a dialógusokat leíró állapotrekordok és szövegrekordok alapján. Az AIR program a DISTAR-B mikroprogramok segítségével kommunikál a memóriában lévő dialógus adatokkal.

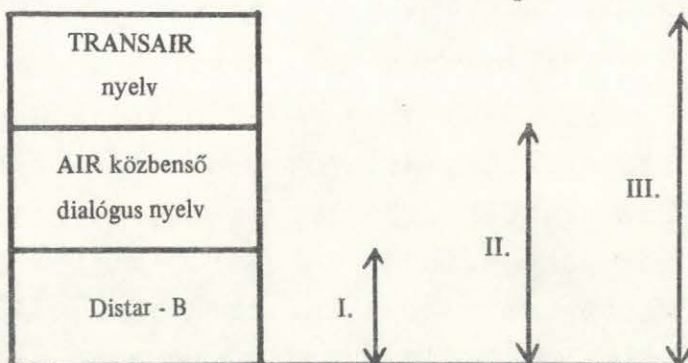
Az interaktív alkatrészprogramíró rendszer felosztható a 9. ábrán látható három részre.



9. ábra

A legalsó szinten helyezkedik el a DISTAR-B általános dialógus rendszer. A DISTAR-B rendszer alapelvével és néhány szubrutinjával egy módszert ad különböző típusu dialógus rendszerek létrehozására. A rendszer önállóan nem használható, de nagyon széles körben alkalmazható. A 9. ábra II. szintje egy szinttel magasabb, mint a DISTAR-B. A második szint valósítja meg egy adott feladatcsoportra a dialógus feltételét, rendszerünkben az APT típusu nyelveken írt alkatrészprogramírást. Összehasonlítva az I. szinttel, nem annyira általános, de már olyan rendszer, amely dialógust képes végrehajtani. A legmagasabb szint a III., már egy konkrét feladatra valósítja meg a dialógust. Minden egyes konkrét feladathoz más-más állapotrekord és szövegrekord tábla tartozik. Külön állapot- és szövegrekord tábla szükséges az EXAPT1-hez, a 2CL-hez stb.

Az állapot és szövegtáblák összeállítására létrehozott nyelvek három szintet alkotnak. A 10. ábrán látható I. szint a DISTAR-B általános dialógus rendszer által ajánlott koncepció, direkt kódu nyelv megírására. Az ajánlott koncepció a felhasználók számára nagyon sok lehetőséget enged meg, így nagyon széles körben használható. Az AIR közbenső dialógus nyelv már figyelembe veszi az alkatrészprogramok tulajdonságait és megadja a lehetőséget arra, hogy adott APT típusu nyelvre leirjuk a dialógusokat.



10. ábra

E direkt kódu nyelv használata nagyon nehézkes.

A TRANSAIR nyelv magas szintű nyelv, amelynek segítségével APT típusu nyelvek dialógusai leírhatók. A 10. ábra legalsó szintjétől felfelé haladva az egyes nyelvek vesztenek általános tulajdonságaikból, de megkönnyítik az adott feladat megoldásának leírását.

Az interaktív alkatrészprogramíró rendszerben felhasznált DISTAR-B általános dialógus rendszer Forgács Tamás és Krammer Gergely munkája [40]. Annak ellenére, hogy a hivatkozott irodalom részletesen foglalkozik a DINFAZ-B általános dialógus rendszerrel, az egész rendszer teljes megértése érdekében röviden összefoglaljuk.

11.1 A DISTAR-B általános dialógus rendszer

A DISTAR-B általános dialógus program jellemzője, hogy egy elemi dialógus a következő lépésekből áll:

- kérdés
- válasz beolvasása
- válasz elemzése
- új állapot meghatározása.

Több elemi dialógus természet szerinti egymásutánja a rendszer. A feladatok sokrétűsége miatt az ilyen rendszerek szerzői általános rendszert nem tudnak javasolni. Ehelyett a következő elemeket tartalmazó számítógépes dialógust ajánlják:

- Egy dialógus program felépítési koncepciója: szövegtábla, állapotrekord, dialógus program.
- Szövegtábla és állapotrekordok gépi tárolásának koncepciója.
- Szövegtábla és állapotrekordok elemeit kezelő, valamint a dialógust végrehajtó általánosan megfogalmazható szubrutinok.

- Szövegtábla és állapotrekordok megadásának külső direkt kódrendszere és az ehhez tartozó loader program.

A DISTAR-B rendszerben a dialógusban szereplő szövegeket előre meg kell adni, azokat a dialógus program egy szövegtáblában tárolja. A dialógus menetét a dialógus állapotainak egymásutánját állapotrekordokkal kell leírni. Az állapotrekordokat a program egy állapottáblában tárolja. Tekintve, hogy általános esetben mind a szövegtábla elemei, a szövegrekordok, mind az állapotrekordok változó hosszúságúak lehetnek, a program tárol egy szövegvektort és egy állapotvektort. Ezen vektorok i -ik eleme az i -ik szöveg, illetve i -ik állapotrekord elemének szöveg, illetve állapottáblabeli indexe.

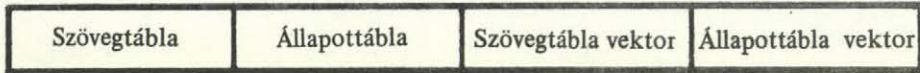
A dialógust leíró szövegtáblát és állapotrekordokat egy loader program (DINFAZ) tölti be a memóriába, amelyet egy ITAB nevű tömb képvisel.

Az ITAB tömb (11. ábra) első részében a szövegrekordok helyezkednek el, ezt az állapotrekordok követik, majd a szövegtábla vektor, illetve az állapottábla vektor. Az utóbbiak a k , illetve l eleme megadja a k szöveg, illetve állapotrekord ITAB-beli helyét.

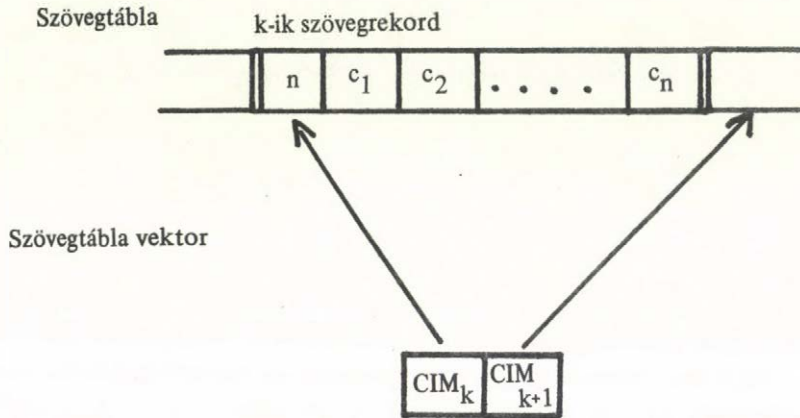
Egy szövegrekord n $c_1 c_2 \dots c_n$ alakú. Az elől álló n egész szám a szöveg karaktereinek száma, ezt követik maguk a karakterek.

Az állapotrekordok két fő részből állnak: akciós rész és elemző rész.

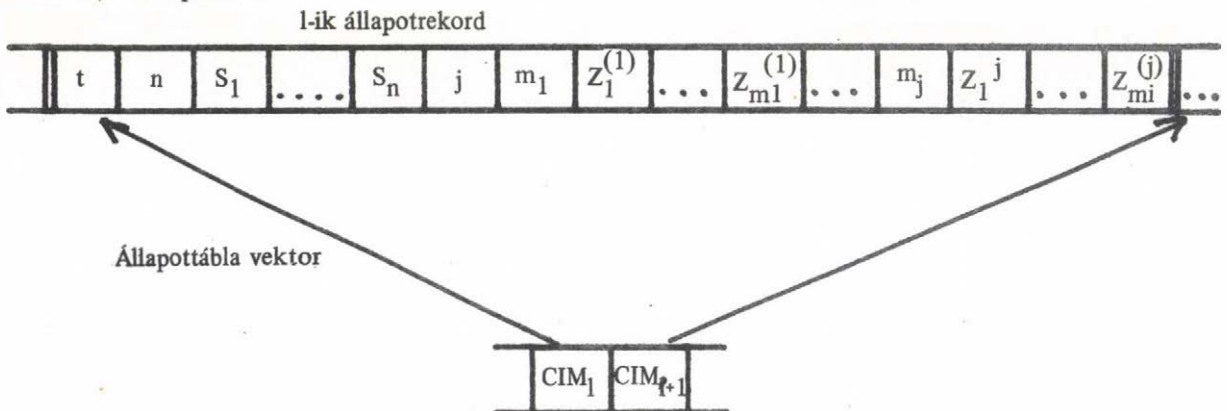
a.) ITAB



b.) Szövegtábla



c.) Állapottábla:



11. ábra

Az akciós rész tárolja egyrészt az un. állapot kezdő akciójára vonatkozó információkat, másrészt a kiinduló kérdéssel kapcsolatos szövegtábla mutatókat, esetleg cursor címeket az alfanumerikus display-n. Az akciós rész tárolási módja:

$$s_1 s_2 \dots s_n$$

ahol n pozitív egész szám az akciós rész hosszát jelöli

s_i az akciós rész i -ik eleme.

Az akciós rész a válaszelemzést és az új állapotki-jelölést vezérli, ezért többnyire szövegtábla mutatókból áll, amelyek az egyes válaszalternatívákhoz tartozó új állapotokra mutatnak.

Az elemző részt újabb egységekre, intervallumokra osztjuk fel. A válaszelemzés minden lépéséhez tartozik egy intervallum. Tárolási módja a következő:

$$j m_1 s_1^{(1)} \dots s_m^{(i)} \dots m_j s_1^{(j)} \dots s_m^{(j)}$$

ahol j az intervallumok száma

m_i az i -ik intervallum hossza

$s_i^{(l)}$ az l -ik intervallum i -ik eleme.

A dialógus szövegrekordjainak és állapotrekordjainak megadására szolgál a szigorú formátu direkt kód.

A program DIAL1 sorral kezdődik és DIAL9 sorral ér véget. DIAL2 jelzi a szövegrekordok kezdetét, amelyet a szövegrekordok követnek. Az állapotrekord tábla kezdetét DIAL3 sor jelzi. Minden állapotrekord új sorban kezdődik.

A szövegrekordok alakja:

$$n \ H \ c_1 c_2 \dots c_n$$

ahol n pozitív egész szám a szöveg karaktereinek száma

H elválasztó jel

c_i i -ik karakter.

Az állapotrekordok alakja:

$$n \ A \ s_1 \dots s_n \ j \ E \ m_1 \ I \ Z_1^{(1)} \dots Z_{m_1}^{(1)} \dots m_j \ I \ Z_1^{(j)} \dots Z_{m_j}^{(j)}$$

ahol n akciós rész hossza

A elválasztó jel

s_i n darab szövegsorszám vagy más paraméter

j intervallumok száma

E elválasztó jel

m_i i -ik intervallum hossza

$Z_k^{(i)}$ i -ik intervallum n -ik eleme.

A DINFAZ szubrutin a külső direkt kódban leírt szavakat beolvassa, majd elválasztó DIAL direktíváknak és a rekordokban szereplő H , A , E , I elválasztó jeleknek megfelelően elhelyezi az ITAB tömbben.

A DIAL a sorok hatására tárolja a szövegtábla és az állapottábla kezdő címét. DIAL 2 és DIAL 3 részek váltakozhatnak DIAL1 és DIAL9 között.

Nagyméretű dialógusok esetén a szövegtábla és állapottábla nem fér be az operatív memóriába. Ilyenkor a dialógus programot szegmentálni kell valamilyen segédtároló felhasználásával. Egy-egy szegmens leírási módja megegyezik a fent leírtakkal. Minden szegmens önálló egységet képvisel. Egyes szegmensek kialakítása három új direktíva bevezetésével tör-

ténik. Minden szegmens elejére kerül egy

§LIM n direktiva

ahol n egész szám a memória igény a szegmens számára

és a § SEGM m n direktiva

ahol m a szegmens sorszáma

n munkamező nagysága.

A dialógus végét egy § END direktívával kell lezárni.

A szegmentált dialógus rendszernél lehetőség van COMMON szövegszegmensek kialakítására is. Ebbe a szegmensbe kerülnek azok a szövegrekordok, amelyekre nagyon sok más szegmens állapotrekordja hivatkozik.

A szegmensek közti kapcsolatot az un. csatoló állapotrekord teremti meg, amelynek formája a következő

02 A - sss nnnn OE

ahol 02 akciós rész száma

A elválasztó jel

-sss negatív egész szám a csatolandó szegmens száma

nnnn a szegmensen belüli állapotrekord sorszáma

O intervallumok száma

E elválasztó jel

A DISTAR-B rendszer egy szubrutin csomagot tartalmaz, amelynek tagjai az ITAB tömb elemeit kezelik. Ismertetőnkben csak azokra térünk ki, amelyeket az interaktív alkatrészprogramíró rendszer felhasznál.

1. J = JAKP (IR, N, L) függvény.

A függvény az állapottábla vektoron megkeresi az IR sorszámú állapotrekord címét. Ennek alapján megkeresi az akciós rész N-ik elemét és annak tartalmát adja a J.

L az akciós rész elemeinek száma.

2. J = JELP (IR, INT, N, L)

A függvény, azaz J értéke megadja az IR állapotrekord INT intervallumának N-ik elemének tartalmát. L az intervallum hossza.

3. J = NER (IR)

A függvény J értéke az IR állapotrekord intervallumainak számát adja meg.

4. SUBROUTINE MOVES (IPUF, I, ISP)

A szubrutin az ISP formál paraméter előjele alapján a megfelelő szövegtáblából kikeresi az ISP-ik szöveget és bemásolja az IPUF tömb I-ik elemétől kezdődően. Az I-t az első szabad memóriaszóra állítja.

5. J = ICSAT (IR)

A függvény a csatoló állapotrekordot kezeli. Megvizsgálja az IR állapotrekord akciós részének első elemét. Ha annak előjele pozitív, J értéke az IR értéket kapja. Ha az akciós rész első eleme negatív, akkor csatoló állapotrekordról van szó, így az első elem abszolút értékének megfelelő szegmenst behívja az operatív memóriába, majd az ICSAT függvény értéke az akciós rész második elemének értékét veszi fel.

6. J = ISUBR (IR)

A függvény az IR állapotrekordot az akciós rész első eleme alapján megvizsgálja, hogy szubrutin lehívó állapotrekordról van-e szó. Ha az elem értéke nem 0.

11.2 AIR közbenső dialógus nyelv

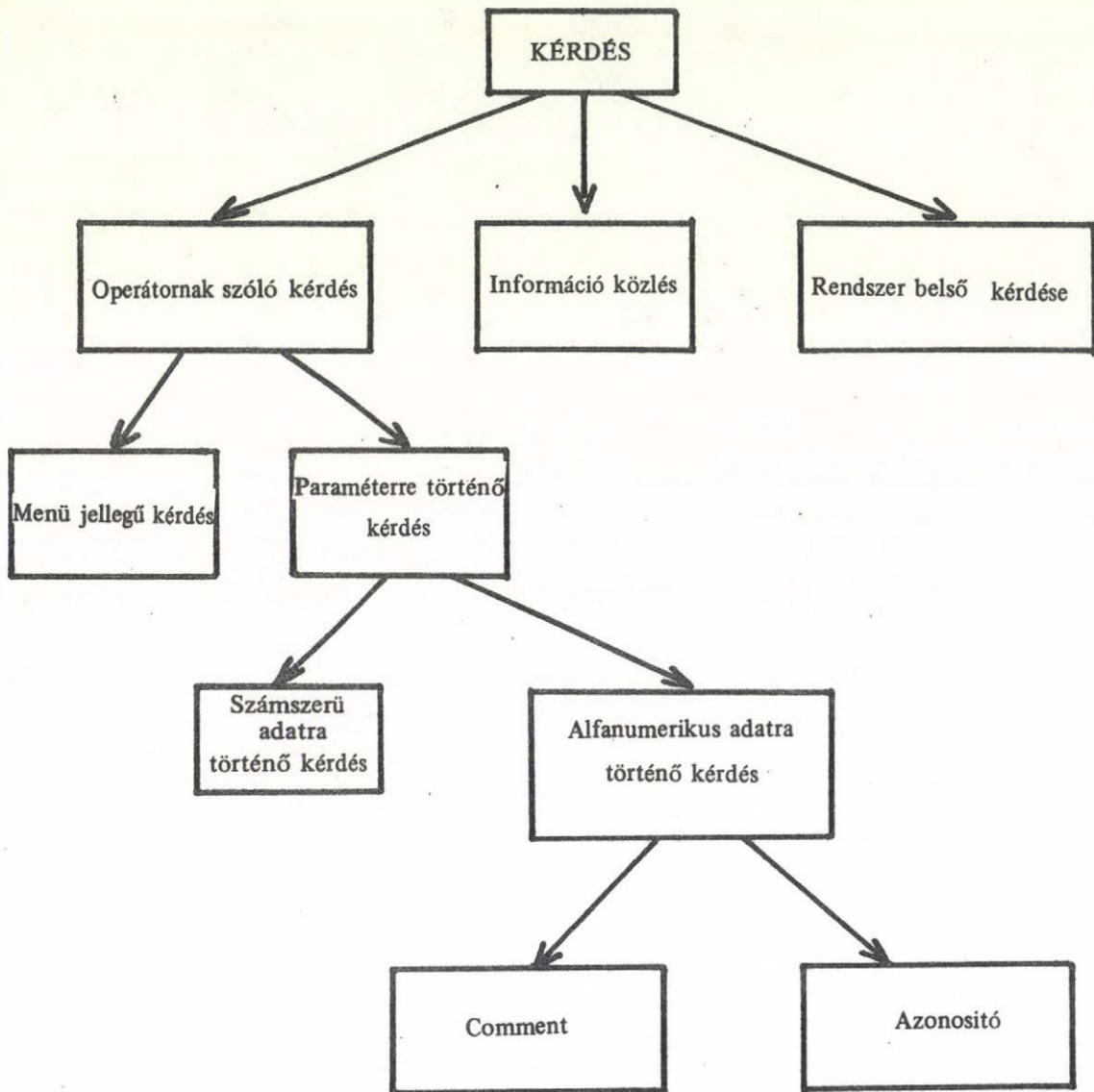
A 8. ábrán bemutatott blokksémában az AIR program hajtja végre a dialógust és állítja össze az APT típusu utasításokat az AIR közbenső dialógus nyelven megírt program alapján. Mielőtt az AIR program kialakításával foglalkoznánk, először az AIR közbenső dialógus nyelvet ismertetjük.

Az AIR közbenső dialógus nyelv megadja azt a lehetőséget, hogy az interaktív alkatrészprogram írásánál felmerülő összes követelményeket kielégítsük, valamint az egyes dialógusokat egyértelműen írjuk le.

Egy dialógust három fő részből építhetünk fel. A három fő rész: a kérdés, amely új dialógust indít el, a válasz eldönti a továbbhaladás útját, a választ analizáló rész, amely a továbbhaladást végrehajtja.

Az alkatrészprogramíró rendszerben a kérdés jellegű információknak három fő típusát különböztetjük meg.

A 12. ábra mutatja a kérdéstípusok fő csoportjait. Az első az operátornak szóló kérdéscsoport, amely kétféle lehet. Az egyik, a menü jellegű kérdéscsoport, jellegénél fogva egyszerre több kérdést tartalmaz. A másik valamilyen paraméterre vonatkozó kérdés. Ez szintén két típusra osztható aszerint, hogy szám-



12. ábra

szerű adatra vagy alfanumerikus információra kérde-
zünk. Az alfanumerikus információk vagy commentek,
vagy azonosítók.

A kérdéstípusok második részébe az ugynevezett
információközlő kérdéscsoportokat soroljuk. Ez a
csoport jellegénél fogva olyan, hogy valamilyen in-
formációt közöl az operátorral (pl. hibajelzés), a
dialógust egy új állapotba viszi anélkül, hogy az
operátortól választ várna vagy kapna.

A harmadik csoportba a rendszer belső kérdéseit so-
roljuk. A belső kérdés anélkül, hogy kívülről vá-
laszt kapna, akciót hajt végre (utasítás összeállí-
tása), majd az akció végrehajtása után új állapotba
viszi a rendszert.

A kérdések fent felsorolt csoportosítása lehetőséget
ad arra, hogy az egyes csoportokba tartozó kérdések-
re külön állapotrekord típusokat alakítsunk ki. Az
operátornak szóló kérdéscsoportba tartozó menü jel-
legű kérdések és a paraméterekre történő kérdéscso-
portok különbözőek, ezért ezekre külön-külön álla-
potrekord típusokat alakítottunk ki.

A fenti megfontolások alapján a következő állapot-
rekord típusokat hoztuk létre.

1. 1. típus: paraméterekre történő kérdéseket
tartalmaz. Ennek három altípusa van (lásd a
12. ábrát)
 1. Számszerű adatra történő kérdés (kód-
száma 10)

2. Azonosítóra történő kérdés
(kódszáma 6)
3. Comment szövegre történő kérdés
(kódszáma 62)

2. 2. típus menü jellegű kérdések
3. 3. típus információ közlések
4. 4. típus rendszer belső kérdései.

Az állapotrekord típusok rögzítése után meghatározzuk, hogy az egyes állapotrekord típusokhoz milyen információkat rendeljünk és azokat hogyan építsük be a nyelvbe.

A kérdés jellegű információkat néhány egyéb információ mellett az állapotrekord akciós részébe, a válaszhoz vagy a válaszokhoz tartozó információkat az elemző rész intervallumaiba helyezzük el.

Az állapotrekordok típusától függően az akciós résznek a következő információkat kell tartalmaznia:

1. Állapotrekordok tipusszáma
2. Állapotrekordok altipusszáma
3. A képernyőre kiírandó szövegek, illetve a rájuk jellemző szövegtábla sorszámok
4. A kiírandó szövegek kezdő karaktereinek cursor címei
5. A várt válasz kezdő karakterének cursor címe.

Az elemző résznek a következő információkat kell tartalmaznia, szintén az állapotrekordoktól függően:

1. Az operátor által adott válasz cursor címét vagy a válaszra jellemző kódszámot

2. A következő állapotokra jellemző állapotrekord sorszámokat
3. Az AIR programban szereplő kapcsoló GOTO utasítás jelzőszámot. (Lásd a 11.4 fejezetet.)

Az egyes állapotrekord típusokat a következőképpen alakítottuk ki.

1. Első típusu állapotrekord

Az akciós rész a következőképpen épül fel:

aa	A
----	---

T	n	SP ₁	CC ₁	SP ₂	CC ₂	SP _n	CC _n	VC	AT
---	---	-----------------	-----------------	-----------------	-----------------	------	-----------------	-----------------	----	----

13. ábra

ahol:

aa egész szám, az akciós részben szereplő információk számát adja meg

T = 1 állapotrekord típusa

n A display-re kiírandó szövegek száma

SP szövegtábla pointer (sorszám)

CC Cursor cím, a kiírandó szöveg kezdő karakterének helye a képernyőn

VC A várt válasz cursor címe

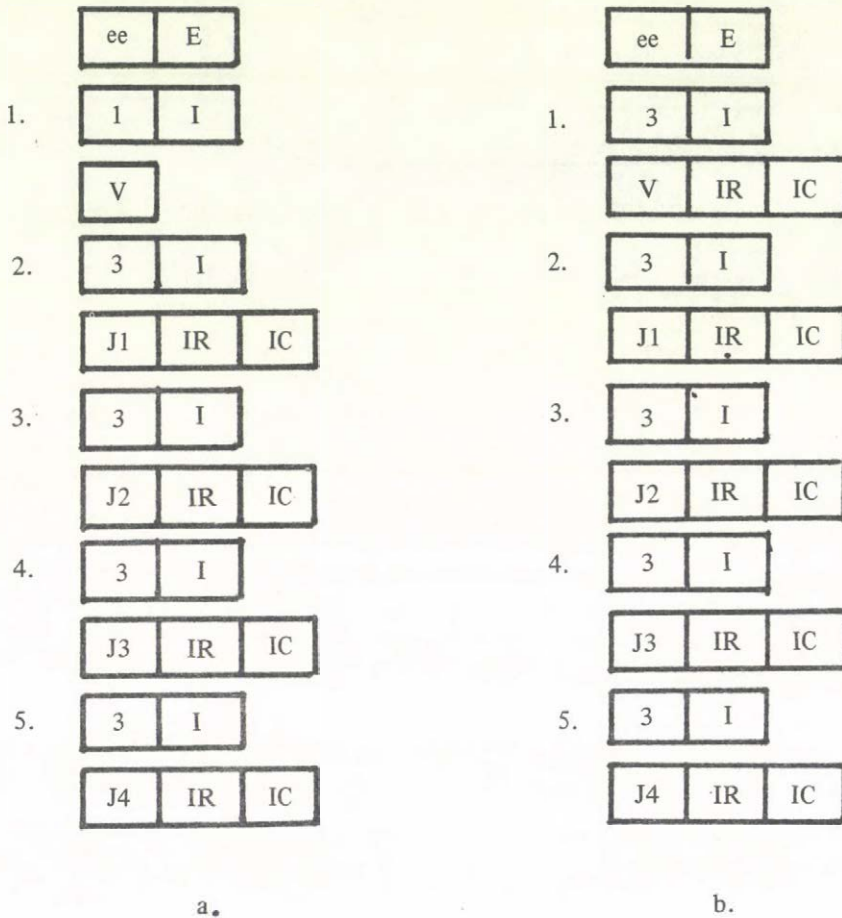
AT Állapotrekord altípus száma. Három értéket vehet fel

AT = 6 ha a kérdés azonosításra vonatkozik

AT = 10 ha a kérdés számszerű információra vonatkozik

AT = 62 ha a kérdés commentre vonatkozik

Két típusu elemző részt különböztetünk meg attól függően, hogy az operátornak engedélyezzük-e a dialóguson belüli visszaugrást vagy sem. A két típusnak megfelelő elemző rész felépítését a 14. ábra mutatja.



14. ábra

A 14/a ábra azt az esetet mutatja, mikor az operátornak nem engedjük meg a dialóguson belüli visszaugrást, a 14/b ábra pedig azt az esetet ábrázolja, amikor a visszaugrás megengedett. Az ábrán az E és I értéket nem magyarázzuk (Lásd 11.1 fejezetet.)

A további karakterek jelentése:

- V a vissza rámutatás cursor címének sora
 "a" esetben V = 0
 "b" esetben V = 14 (14 sor)
- J1 = 1 kódszám arra az esetre, amikor az operátor a kijelölt helyre nem adja meg a választ
- J2 = 2 kódszám arra az esetre, ha az operátor által megadott azonosító számmal kezdődik
- J3 = 3 kódszám arra az esetre, ha az operátor által megadott azonosító diagnosztikailag hibás
- J4 = 4 kódszám arra az esetre, ha a megadott információ hibátlan
- IR a következő dialógust leíró állapotrekord sorszáma
- IC kapcsoló GOTO utasítás jelzőszáma.

AT = 10 és AT = 62 altípusú állapotrekordok esetén a 14. ábrán feltüntetett intervallumok közül a 3. és 4. intervallum értelmét veszti, így elmaradhat.

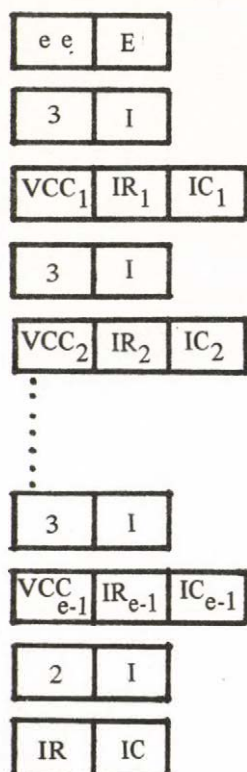
2. Második típusú állapotrekord

E típusú állapotrekorddal írjuk le a menü jellegű dialógusokat. Egy dialógus több kérdésből áll és a választ rámutatással kell megadni. A rendszert könnyebb és rugalmasabb kezelhetőség miatt úgy alakítottuk ki, hogy a cursor rámutatás ne egy megadott karakterre történjék, hanem a kérdést tartalmazó sorra. Mivel a menü több kérdést tartalmaz, ezért a dialógus megtervezésekor úgy alakítottuk ki az egyes kérdéseket, hogy azok külön-külön sorban helyezkedjenek el.

A második típusu állapotrekord akciós része megegyezik az első típusu állapotrekord akciós részével, két paraméter értelemszerű megváltoztatásával. A 13. ábrán szereplő T értéke 2 lesz, AT értéke pedig 1, mivel csak egy altípusa van az állapotrekordnak. VC-nek csak egy értéket adhatunk annak ellenére, hogy több adandó válasz lehetséges. Célszerű azt a cursor címet beírni, amely a leggyakrabban előforduló válaszhoz tartozik.

A második típusu állapotrekord elemző részét mutatja a 15. ábra.

A 15. ábrában szereplő karakterek jelentése a következő:



15. ábra

VCC n válasz cursor cím

A menü egyes elemeire adandó válaszhoz
tartozó képernyősor sorszáma

IR a válasznak megfelelő következő dialó-
gust leíró állapotrekord sorszáma

IC kapcsoló GOTO-utasításra jellemző szám.

Az elemző rész utolsó intervalluma két elem-
ből áll. Ez arra a célra szolgál, hogy hibás
rámutatás esetén, azaz, ha olyan sorra tör-
tént a rámutatás, amelyre az állapotrekord nem
készült fel, más dialógusba tudjuk vinni a
rendszert (hibajelzés stb).

3. Harmadik típusu állapotrekord

E típusu állapotrekordba soroltuk azokat a
kérdéstípusokat, amelyek kérdéseket vagy in-
formációkat közölnek az operátorral a display-n
keresztül. A dialógus a kérdésekre választ nem
vár, hanem válasz nélkül egy újabb állapotba
viszi a rendszert.

Az állapotrekord akciós részét mutatja a 16.
ábra.

aa	A
----	---

T	n	SP ₁	CC ₁	SP ₂	CC ₂	...	SP _n	CC _n	VC
---	---	-----------------	-----------------	-----------------	-----------------	-----	-----------------	-----------------	----

16. ábra

A 16. ábrán szereplő paraméterek a 13. ábrán feltüntetett paraméterekkel egyeznek meg, kivéve a VC és T értékét. Itt $T = 3$ értéket vesz fel.

VC A következő dialóguskérdés kezdő, vagy hibakiírás esetén, a hiba korrigálásának kezdő karakterhely cursor címe.

Az állapotrekord elemező része egy intervallumból áll, amely két elemből tevődik össze. Az első eleme a következő dialógust leíró állapotrekord sorszáma, a második eleme a AIR programban szereplő kapcsoló GOTO-utasításra jellemző szám.

4. Negyedik típusu állapotrekord

Az ilyen típusu állapotrekordok közé soroljuk azokat a kérdéstípusokat, amelyek az operátortól választ nem várnak, valamilyen akciót végrehajtanak és egy új dialógusba viszik át a rendszert.

Az állapotrekordok akciós részének kell tartalmaznia mindazt az információt, amely az akció végrehajtásához szükséges. Rendszerünkben az akció végrehajtása az operátor által kiválasztott APT típusu utasítás összeállítását jelenti, amely szótári szavakból, azaz szövegtáblában elhelyezett szavakból és az operátor által megadott paraméterekből tevődik össze. Az akciós rész felépítését a 17. ábra mutatja.

aa	A
----	---

T	n	SP ₁	SP ₂	9999	SP _i	...	9999	SP _n	AT
---	---	-----------------	-----------------	------	-----------------	-----	------	-----------------	----

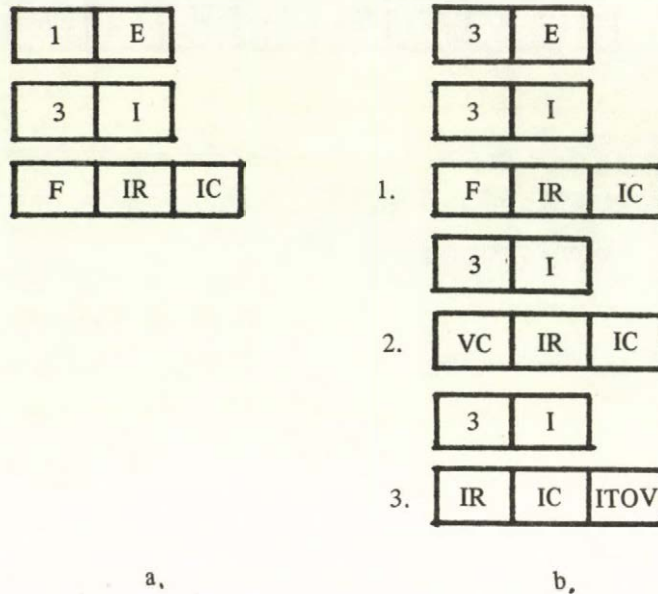
17. ábra

Az ábrában szereplő paraméterek jelentése a következő:

- T = 4 az állapotrekord típusa
- n az összeállítandó APT típusu utasításban szereplő szavak és az operátor által megadott paraméterek száma
- SP szövegtáblában elhelyezett APT típusu szótári szavak szövegtáblabeli sorszáma
- 9999 Az operátor által megadott paraméter
- AT = 1 állapotrekord altípus száma.

Egy APT típusu utasítás felépítéséhez a rendszer több kérdést tesz fel az operátornak. Minden paraméter megadása után egy 4-es típusu állapotrekorddal építjük be a választ az APT típusu utasításba. Ha egy APT típusu utasítás felépítése még nem fejeződött be, újabb kérdéssel kell fordulni az operátorhoz. Viszont, ha az utasítás felépítése már befejeződött, az összeállított kész utasítást ki kell írni a képernyőre végső ellenőrzés céljából és jóvá kell hagyni az operátorral. Ha a megjelenített utasítást az operátor nem fogadja el, vissza rámutatással törli, egyéb sorra történő rámutatás esetén elfogadja.

E kettős feladatot az elemző rész kettős kialakításával oldottuk meg. Ezt mutatja a 18. ábra.



18. ábra

A 18/a ábra mutatja azt az esetet, amikor még az utasítás felépítése nem fejeződött be.

A 18/b ábrán látható az az eset, amikor az utasítás felépítése már befejeződött.

Az ábrán feltüntetett jelölések:

F jelző szám (flag), jelentése az, hogy az utasítás felépítése befejeződött-e már vagy sem.

F = 0 az utasítás felépítése befejeződött

F = 1 az utasítás felépítése nem fejeződött be

VC vissza-rámutatás cursor címe

IR új állapotrekord sorszáma

ITOV alkatrészprogram befejezését mutató flag, egyben kapcsoló GOTO-ra jellemző szám.

Ha ITOV = 1 alkatrészprogramírás nem fejeződött be

ITOV = 2 alkatrészprogramírás befejeződött.

A 18/b ábrán látható elemző rész 1. intervallumában szereplő IR érték helyére az aktuális állapotrekord sorszámot kell beírni. Az elemző rész második intervalluma azt az esetet tartalmazza, amikor az operátor a megjelentetett utasítást nem fogadja el. A harmadik intervallumban szereplő IR annak az állapotrekordnak a sorszáma, ahol a dialógust folytatni kell, ha az operátor az összeállított APT típusu utasítást elfogadja.

11.3 A TRANSAIR az állapotrekordokat leképező rendszer

Egy APT típusu nyelvre kialakított interaktív alkatrészprogramíró rendszer nagy mennyiségű dialógusból áll. A nagy mennyiségű dialógust (EXAPT1 8000 db, 2 CL 9500 db dialógus) AIR közbelső dialógus nyelven leírni nagyon fárasztó és nagy hibalehetőséget involváló munka. Az állapotrekordok kézi megírása, annak lyukkártyára vagy lyukszalagra lyukasztása több hónapot igényel. Az így elkészített állapotrekordokból a hibák kiszűrése további több hónapos intenzív munka, amelyet még a számítógép futtatási költségei is terhelnek.

E munka tökéletes elvégzését megnehezíti a nagyon sok és szerteágazó dialógus utak végigjárása.

A felsorolt nehézségek megoldására hoztuk létre a TRANSAIR-t az állapotrekordokat leképező rendszert.

11.31 A TRANSAIR nyelv kialakításának szempontjai

A TRANSAIR dialógus nyelv megtervezésénél fő szempontunk az volt, hogy bármely APT típusu nyelvre már előzőleg megtervezett és elkészített dialógusokat könnyen lehessen leírni és az e nyelven megírt programokat könnyen és gyorsan lehessen javítani. Ezt a célt jó hatásfokkal úgy tudjuk elérni, hogy bizonyos kötöttségeket építünk be a rendszerbe a dialógustervezés szabadságának rovására. Általában nem lehet a rendszerben változtatni ezeket a kötöttségeket, azaz felszabadításukra nincs lehetőség, de bizonyos korlátok rugalmasan változtathatók, illetve módosíthatók. A beépített megszorítások az előző fejezetekben tárgyalt elvekkel szoros összhangban vannak.

A kötöttségek a következők:

1. Az operátornak a feltett kérdésre a választ a képernyő jobboldalán kell megadni. A válaszok helyét kijelölő marker jelek a képernyő jobboldalán jelennek meg. Kiírásukról a rendszer gondoskodik.
2. A választípusoktól függően a két marker jel közti távolság kötött. Azonosító válasznál 6, számszerű válasznál 10, comment válasznál 62 és rámutatás esetén 1 karakterhely.
3. A display képernyőn egyszerre csak egy új kérdés jeleníthető meg. Újabb kérdés csak a válasz beérkezése után jelenik meg.

- A menü, jellegénél fogva kivétel ez alól.
4. A dialógus-fán történő visszalépést biztosító rámutatás-hely a display képernyő utolsó előtti során helyezkedik el (14 sor). A dialógus tervezőjének erről külön nem kell gondoskodnia.
 5. A rendszer a megjegyzéseit, illetve közlő információit a képernyő utolsó sorába helyezi el (15 sor).
 6. Rámutatásnál elegendő csak a képernyő megfelelő sorát megjelölni, nem szükséges a kijelölt karakter helyet.

A felsorolt kötöttségek csak első olvasásra tűnnek merevnek. A felsorolt egyes megszorításokat nem a létrehozott nyelv követeli meg, hanem maga az APT típusu nyelv. A többi megszorítást a dialógus ergonómiai megfontolásai írják elő.

Az AIR közbenső dialógus nyelv, számítógépes hasonlattal élve, gépi nyelv. A benne foglalt információk kódolt formában jelennek meg, amelyeket a rendszer írójának be kell tartani. Az általunk javasolt nyelv nagyon egyszerű, mégis magas szintű, erősen probléma-orientált nyelv és az AIR program használatát tételezi fel.

11.32 Az alkatrészprogramíró rendszerben használt dialógusok fajtái és azok felosztása

APT típusu utasítások felépítése, az alkatrészprogramíró rendszerben, dialógusok egymás utáni sorozatával történik. Egy utasítás felépítéséhez tartozó dialógusok egy képernyőképen jelennek meg. Minden új APT típusu utasítás fel-

építése új képernyőképpel kezdődik. Egy APT utasítás felépítéséhez tartozó dialógusokat fődialógusoknak és az utasítás felépítéséhez tartozó egyes dialógusokat aldialógusoknak fogjuk a következőkben nevezni. Egy fődialógussal egy APT típusu utasítást írhatunk le, amely több aldialógusból állhat.

Az alkatrészprogramíró rendszerben használt dialógusokat hét csoportra oszthatjuk a következőképpen:

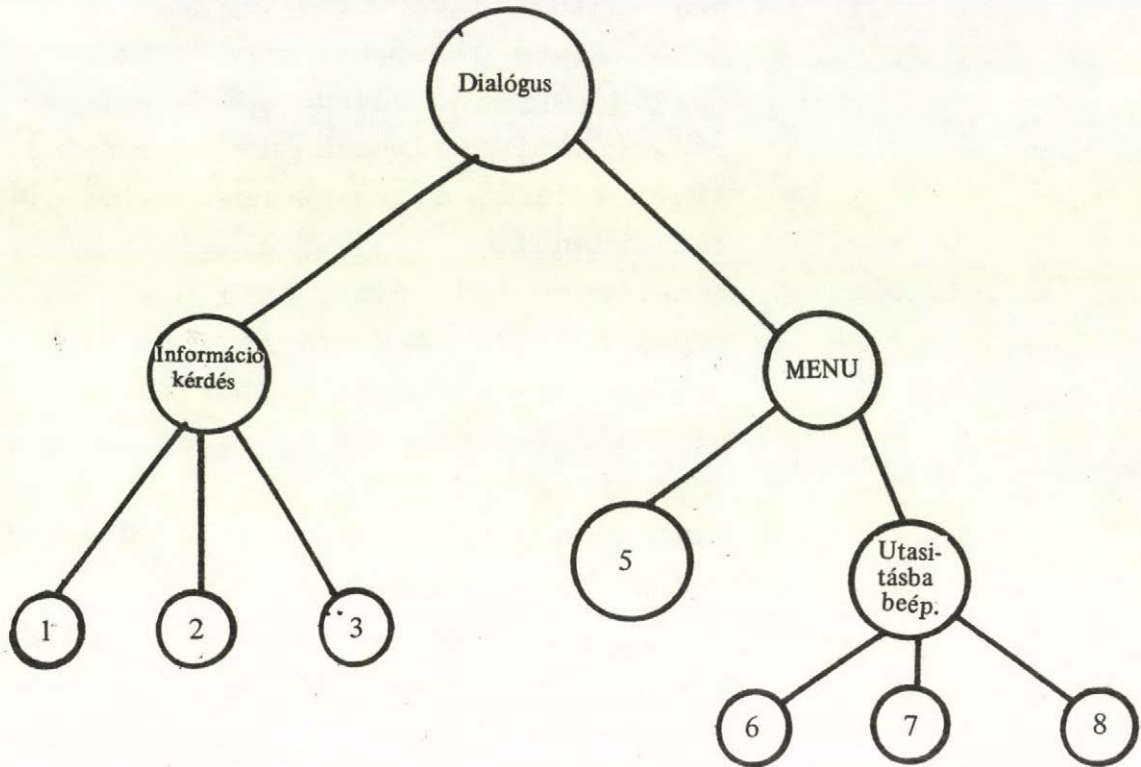
1. A rendszer által feltett kérdésre az operátornak számszerű, azonosító vagy comment választ kell adnia. Ha az operátor a választadást elmulasztja, a rendszernek hibajelzést kell adnia.
2. A rendszer számszerű, azonosító vagy comment választ kér az operátortól, de válaszolni nem szükséges. Ha a választadás megtörténik, a válaszhoz tartozó információt beépíti az épülő APT típusu utasításba, de ha a választadás nem történik meg, új aldialóguson folytatódik a rendszer futása hibajelzés nélkül.
3. A rendszer számszerű, azonosító vagy comment választ kér az operátortól, de válaszolnia nem szükséges. Választadás esetén a válaszhoz tartozó információ beépül az APT típusu utasításba és a következő aldialóguson folytatódik a rendszer futása. Ha nincs válasz, a fődialógus lezárul, azaz az APT típusu utasítást, a fődialógusba eddig befutott információk alapján, a rendszer összeállítja és felkinálja az

operátornak végső ellenőrzésre. A rendszer egy másik fődialogusra ugrik.

4. Menü típusu dialógus, ahol a választól függően egy fődialogusra történik az ugrás. (Utasítástípusok kiválasztása.) Ha nincs válasz, a rendszernek hibajelzést kell kiadnia.
5. Menü típusu dialógus, ahol a választól függő APT utasítás szerinti szótári szót építjük be az aktuális utasításba. Válaszadás elmulasztása hibajelzést kell, hogy eredményezzen.
6. Menü típusu dialógus, ahol a választól függő APT utasítás szerinti szótári szó kerül beépítésre az éppen épülő utasításba. Válaszadás elmulasztása esetén, hibajelzés kiírása nélkül, a következő aldialoguson folytatódik a program futása.
7. Menü típusu dialógus. A képernyőn csak egy kérdés jelenik meg. Az operátor tetszőleges sorra való rámutatással (kivéve a vissza-sort) fogadhatja el a felajánlott kérdést. Elfogadása esetén a hozzátartozó információ beépül az APT típusu utasításba.

Az egyes dialógustípusokat a 19. ábrán feltüntetett fa struktúrával lehet könnyen szétválasztani, illetve rendszerezni.

A fent leírt csoportosítás egyrészt a nyelv könnyű kialakítását tette lehetővé, másrészt a TRANSAIR fordítóprogram megvalósítását egyszerűvé.

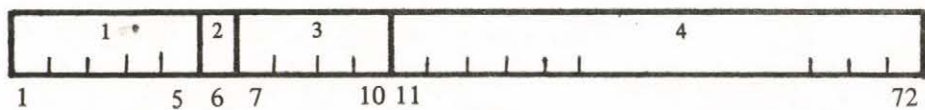


19. ábra

11.33 A TRANSAIR nyelv leírása

A TRANSAIR nyelv a következő szintaktikával rendelkezik.

A program rekordokból épül fel. Egy rekord máximalisan 72 karakterből állhat. Általános alakját a 20. ábra mutatja.



20. ábra

Egy rekord négy mezőből épül fel.

1. A címkemező 5 karakterhelyet foglal el; 1-5 oszlopok.
2. Utasításmező. A 6-os oszlopot, azaz egy karakterhelyet foglal el.
3. Utasítást kiegészítő mező, 4 karakterhelyből áll (7-10 oszlopok).
4. Szövegmező 11-72 oszlopokat foglalja el, összesen 62 karakterhelyet.

A címkemezőbe, valamint az utasítást kiegészítő mezőbe csak számszerű információkat lehet beírni. Az utasításmezőben csak az utasításra jellemző karakter állhat. A szövegmezőben akár számszerű, akár karakterekből álló információ lehet.

Utasítások

1. "A" (Answer) operátortól várt válasz definiálása.

Általános alak:

Aaabb

ahol

aa kétjegyű egész szám. Megadja az operátortól kért válasz display képernyő sorának számát

bb kétjegyű egész szám. Az operátortól várt válasz típusát, azaz a karaktereinek számát adja meg.

bb a következő értékeket veheti fel
6 válasz 6 karakterből áll (azonosító válasz)

10 válasz 10 karakterből áll (számszerű válasz)

62 válasz 62 karakterből áll (comment válasz)

2. "C" (Call) szubrutint lehívó utasítás
Általános alak:

Cccccaaaa

ahol

aaaa a lehívott szubrutin neve. Alfabetikus karakterekből áll. Maximálisan 4 db.

3. "E" (END) szubrutint befejező utasítás
Általános alak:

E

4. "F" (FINI) dialógus programot befejező utasítás

Általános alak:

F

A TRANSAIR nyelven megírt program utolsó utasítása.

5. G (GOTO) ugró utasítás
Általános alak:

Gcccc*aaa

ahol

aaa egy- vagy többjegyű egész szám, a programba bárhová beépített címke száma

6. J (JUMP) APT utasítást kiválasztó menü ugró utasítása

Általános alak:

Jbbcc*aaa

ahol

aaa egy vagy többjegyű egész szám. A programba bárhová beépített címke száma.

bb display képernyő sor száma, ahová a rendszer az operátor rámutatásos választ várja.

7. "M" (MENU) Menü típusu utasítás, ahol a választól függő APT szótári szó kerül beépítésre az éppen épülő utasításba.

Általános alak:

M aa00 cccccc

ahol

aa kétjegyű egész szám, több értéket vehet fel.

- display képernyősor száma, ahová a rendszer az operátor rámutatásos választ várja. A rendszer hibajelzést ír ki, ha nem kap választ az operátortól.
- 00 ha az operátor nem ad megfelelő választ, a rendszer hibajelzés nélkül tovább megy a következő dialógusra. cccccc információnak ez esetben nincs jelentősége.
- 99 a megadott cccccc információ tetszőleges képernyősor rámutatás esetén (kivéve a vissza-sort) beépül az APT utasításba.

ccccc karakterekből és számokból álló információ (maximálisan 62), amely beépül az APT típusu utasításba.

8. "P" (PROGRAM) dialógus program első utasítása.

Általános alak:

P

9. Q (QUESTION) kérdés definiálása

Általános alak:

Qaabbcccccc.... cc

ahol

- aa display képernyősor száma, ahová a cccc-vel jelzett szöveg kerül.
- bb display képernyőoszlop száma, ahová a ccccc-vel jelzett szöveg első karaktere kerül
- cccccc alfanumerikus karakterekből és számokból álló szöveg, amely a képernyőre kerül kiíratásra. Lehet common szöveg sorszám is ϕ -al kezdve.

10. "R" (RESULT) aldialógus eredményét leíró

utasítás

Általános alak:

R aabbcccccc

ahol

- aa több értéket vehet fel
- 00 az operátor által adott választ a ccccc-vel jelzett szöveg után illesztve építi be az APT típusu utasításba.
- 00 ugyanaz mint a "00"
- 01 az operátor által adott választ a ccccc-vel jelzett szöveg elé illesztve építi be az APT típusu utasításba
- bb több értéket vehet fel
- uu ha az operátor a feltett kérdésre nem ad választ, hibajelzést ad a rendszer.
- bb pozitív egész szám. Ha az operátor a feltett kérdésre nem ad választ, a rendszer nem ad hibajelzést. A fődialogust lezárja, a program a bb címkével jelzett fődialogusra ugrik.

--b tetszőleges negatív szám. Ha az operátor a feltett kérdésre nem ad választ, hibajelzés nélkül a következő aldialoguson folytatódik a program futása. Fődialogus végén nem állhat.

11. "S" (SUBROUTINE) szubrutin utasítás

Általános alak:

Scccc aaaa

ahol

aaaa alfanumerikus karakterekből álló szöveg, a szubrutin neve. Maximálisan négy karakterből állhat.

12. "T" (TEXT) utasítás common szövegeknek a szövegtáblába történő beépítésére.

Általános alak:

T aaaa ccccccc

ahol

aaaa szövegtábla sorszáma

cccc maximálisan 62 alfanumerikus karakterből, illetve számokból álló szöveg.

11.34 A TRANSAIR nyelven irt program szabályai

A TRANSAIR nyelven irt program előzőleg megtervezett alkatrészprogram-utasításokat összeállító dialógusok leírására szolgál. Az e nyelven megírt programok szabályait foglaljuk össze.

Minden programot "P" utasítással kell kezdeni és "F" utasítással kell lezárni. A "T" rekordok lehetőséget adnak arra, hogy az állandóan ismétlődő szövegeket a program írása előtt szövegtáblá-

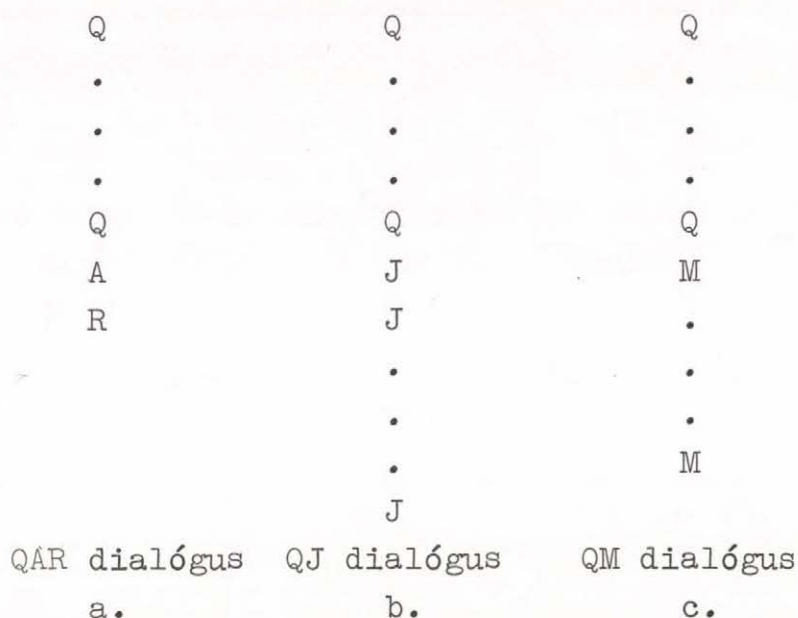
ba helyezzünk el (minden új szöveg egy-egy új rekord) és később annak sorszámaival hivatkozunk rá. A "T" rekordokat a "P" utasítás után kell elhelyezni.

Egy APT utasítást összeállító dialógusok, valamint azok a menük, amelyek segítségével egy kívánt APT típusu utasítást megkeresünk fődialógusokban történik. Minden fődialógus első aldialogusát címkével kell ellátni.

A címkek több funkciót látnak el.

- a TRANSAIR fordítóprogrammal közli, hogy új képet kell előállítani a display képernyőn, azaz új fődialógus kezdődik.
- Lehetővé teszi, hogy egyik fődialógusból egy másik fődialógusra lehessen ugrani.
- Ha a címkézést sequenciálisan végezzük, a címke száma megadja az interaktív rendszerhez tartozó képek, azaz a fődialógusok sorszámát.

Egy aldialogus felépítése kétféle lehet, attól függően, hogy a kérdés valamilyen paraméterre történik-e, vagy a kérdés feltevés menü jelle-gű-e. Ha a kérdés feltevés valamilyen paraméter-re vonatkozik, a 21/a ábrán látható utasításokat kell használnia. Továbbiakban QAR dialógusnak fogjuk nevezni.



21. ábra

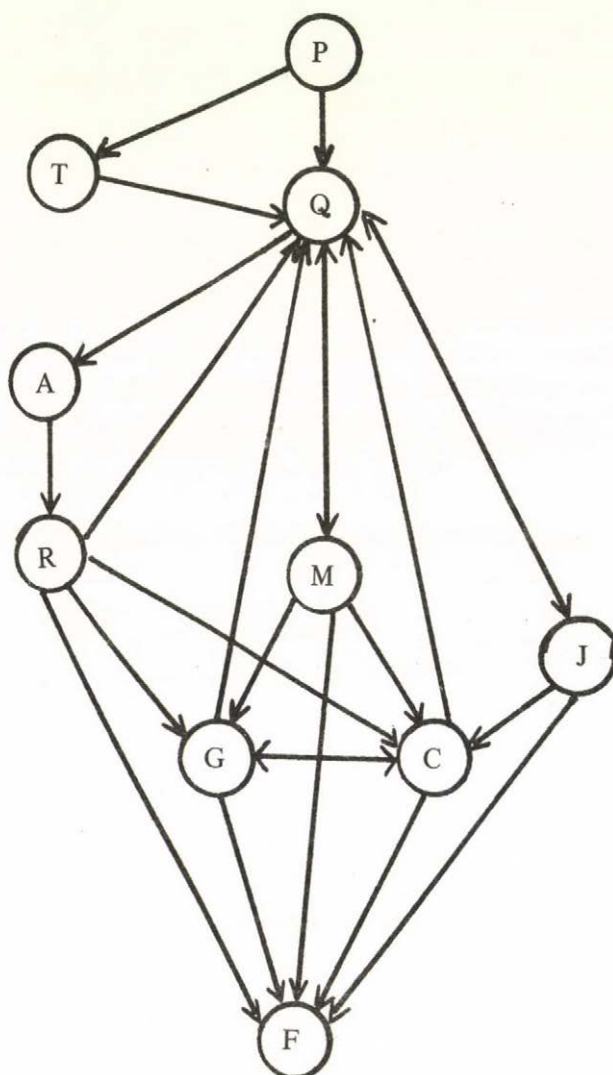
Ha a kérdésfeltevés menü jellegű és APT típusu utasítások kiválasztására szolgál, a 20/b ábrán látható utasítások használhatók (QJ dialógus). Ha pedig az APT típusu utasításba beépülő menüről van szó, a QM dialógust használhatjuk (20/c ábra).

A fentiekből következik, hogy minden aldialógusnak Q-val, azaz kérdéssel kell kezdődnie és R-rel, J-vel vagy M-mel kell befejeződnie. A J-k, illetve az M-ek számát a menü elemek száma határozza meg, kivéve a 6-os dialógus esetet, amikor MOO utasítással 1-el bővül az M-ek száma.

Egy fődialógus egy QJ dialógusból vagy QAR és QM dialógusok tetszőleges sorozatából áll. A fentiek szerint fődialógust egy címke nyit meg és a következő fődialógus címkéje vagy egy G ug-ró utasítás zár le.

Bármely aldialógus helyére egy "C" utasítást is beépíthetünk, amely aktivizálja az általa megnevezett szubrutin dialógust.

A nyelvben definiált utasítások bizonyos fokig egymás függvényei, ezért nem lehet az utasításokat tetszőleges sorrendben írni. Az utasítások egymáshoz fűződő kapcsolatát a 22. ábrán felrajzolt gráf mutatja.



22. ábra

A nyelv lehetővé teszi, hogy egy vagy több aldialogusból álló szubrutin dialógust alakítsunk ki és azt a "C" utasítással tetszőleges helyeken többször lehívjuk. A szubrutinnak "S" utasítással kell kezdődnie és "E" utasítással befejeződnie. Az "S" és "E" utasítások közé csak QAR és QM aldialogusok építhetők be. Nem építhetők be a T, G, J, C és az F utasítások. Az R utasítások közül nem építhető még be R-~~o~~bb típusu utasítás, ahol a bb pozitív egész szám. Szubrutinok egy-
másba nem ágyazhatók. Az "S" és az "E" utasítások között címkét nem szabad használni.

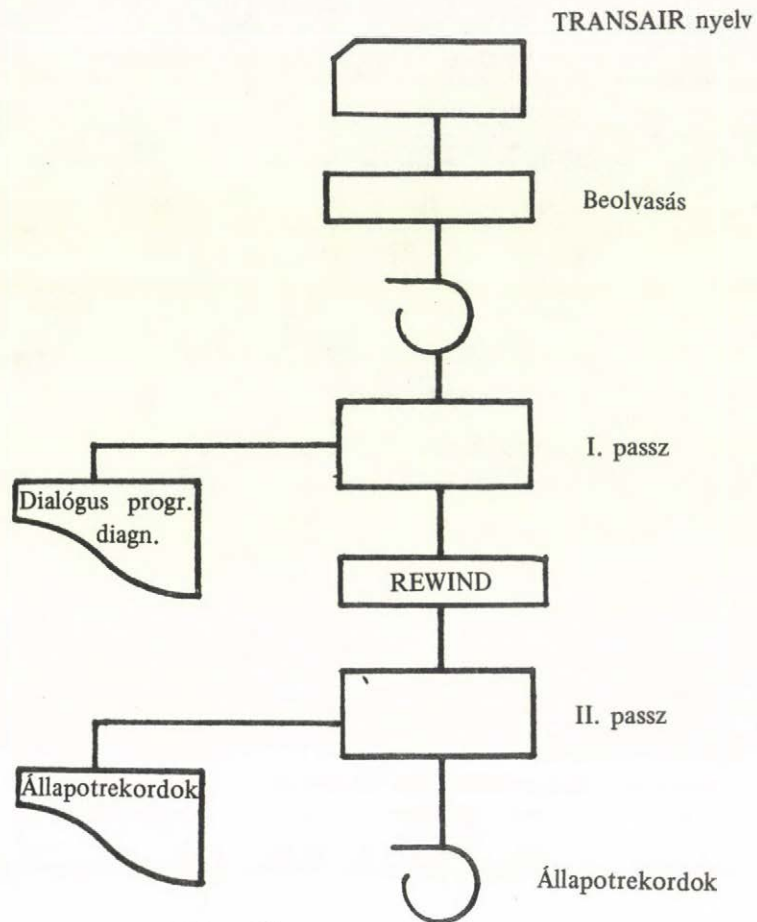
"C" utasítással csak olyan szubrutin hívható, amelynek definiálása a "C" utasítást megelőző utasítások közt már szerepelt.

11.35 A TRANSAIR fordító program

A TRANSAIR fordító program olvassa be és dolgozza fel a TRANSAIR nyelven megírt programot. Inputja a TRANSAIR nyelven megírt program, outputja pedig az AIR közbenső dialógus nyelv formátumú állapotrekordok, illetve szövegtáblák.

A TRANSAIR nyelven megírt program feldolgozása két passzos futásban történik a 23. ábrán látható módon. A lyukkártyákra vagy lyukszalagra lyukasztott programot a beolvasó program egy munka file-ra teszi, hogy a program kétszeri beolvasását gyorsan meg lehessen valósítani. Az I pass programrész a következő funkciókat látja el.

- Rekordok beolvasása
- Szintaktikus ellenőrzés



23. ábra

- Szemantikus ellenőrzés. A diagnosztikai vizsgálatnál a program felhasználja a 22. ábrán látható gráfból alkotott mátrixot, amelyből az egyes utasítások egymáshoz történő összerendelése, illetve azok összerendelését megsértő hibák könnyen leolvashatók.
- T és Q rekordok alapján a szövegtábla összeállítása.
- Második passz futását megkönnyítő táblázat összeállítása, amely a következő adatokat tartalmazza.
 1. Fődialogus sorszáma (címke)
 2. Aldialogusok tipusszáma

3. Tipusszámok alapján az aldialógusokhoz tartozó állapotrekordok száma
4. Aldialógusokhoz tartozó állapotrekord sorszámkiosztása
5. Az ugró utasításokhoz tartozó állapotrekord sorszámkiosztása
 - Lista a dialógus programról
 - Lista a diagnosztika eredményéről.

Az első passz csak akkor engedi a második passz futását, ha a diagnosztikai programrész fatális hibát nem jelzett.

A fatális hibát nem tartalmazó dialógus programot olvassa be a II. passz programrész. Feladata a beolvasott rekordok, valamint az első passz-ban összeállított táblázat alapján az állapotrekordok összeállítása.

A hét dialógustípus mindegyikéhez egy-egy állapotrekord sorozat rendelhető hozzá. A sorozatokhoz tartozó állapotrekordok akciósrészei négy, elemzőrészeinek intervallumai egy Fortran szubrutinnal előállíthatók, ha a szubrutinokhoz tartozó, a dialógustípusoktól függő paramétereket más módon állítjuk elő.

A dialógustípusoktól függő paraméterek előállítását egy táblázat kialakításával oldottuk meg. A táblázat tartalmazza, hogy az egyes dialógustípusokhoz tartozó állapotrekordok összeállítására milyen szubrutin lehívásokat kell elvégezni, milyen sorrendben és azokhoz milyen paraméterek tartoznak. E szervezőtábla adatait egyrészt az első passz által összeállított táblázat,

valamint az újra beolvasott dialógusprogram utasításáiban szereplő paraméterek egészítik ki.

A második passz programrészben összeállított állapotrekordokat a program egy file-ra helyezi és listát készít róluk.

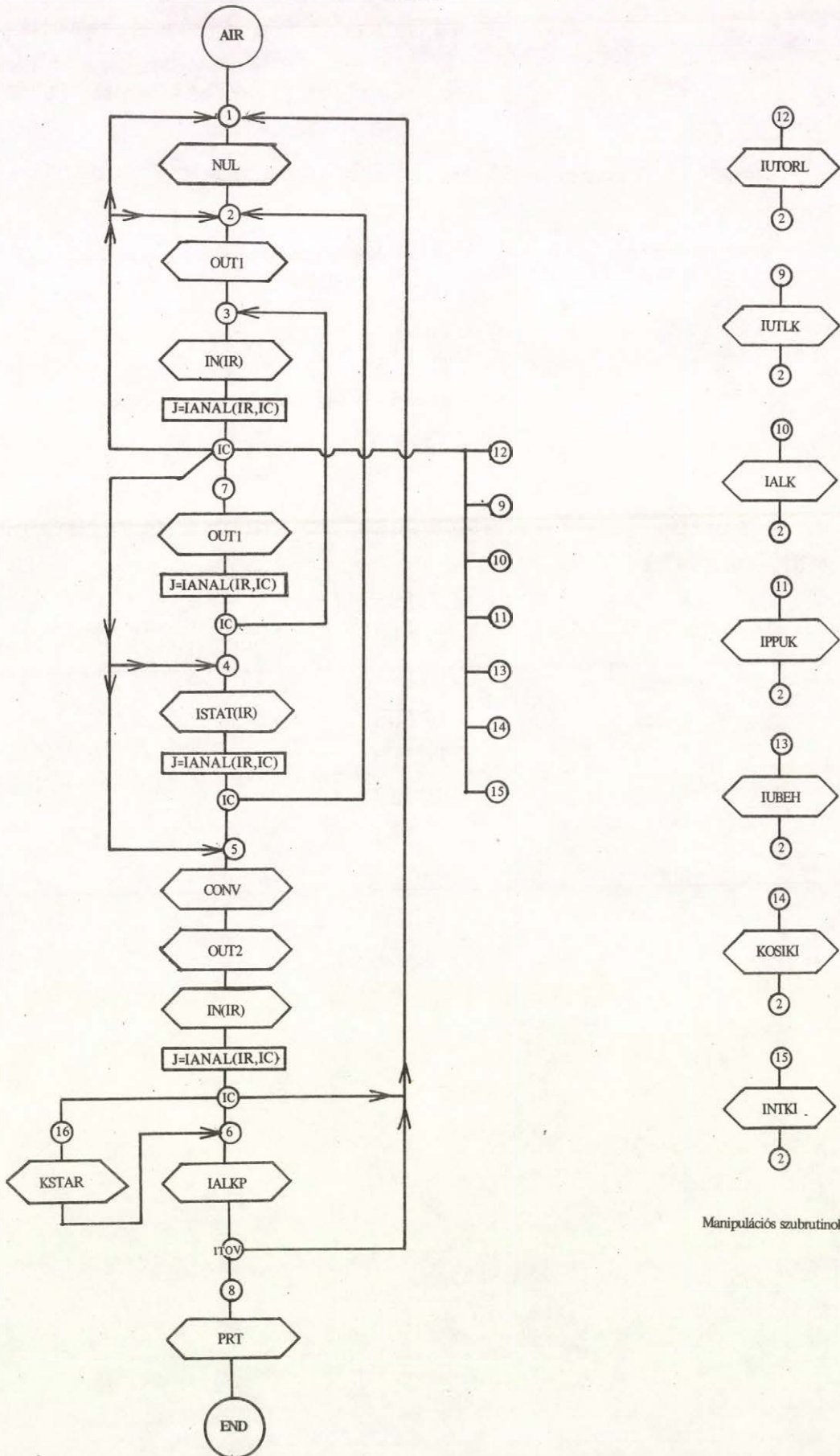
A 3. számú függelék példát mutat be a TRANSAIR dialógus nyelv használatára, valamint bemutatja a TRANSAIR fordítóprogram outputját.

11.4 AIR dialógusprogram

Az operátor és a rendszer közötti dialógusról az AIR közbelső dialógusnyelven írt állapotrekordok alapján az AIR program gondoskodik.

Az AIR program blokksémáját a 24. ábra mutatja. A program szegmentáltan szubrutinokból és függvényekből épül fel. Fő része hét szubrutinból, illetve függvényből áll, a manipulációs lehetőségeket ugyancsak hét szubrutin látja el és az alkatrészprogram végső ellenőrzéséről egy szubrutin gondoskodik. A programozás megkönnyítésére még öt szubrutint alakítottunk ki ismétlődő funkciók ellátására.

A program működésének ismertetése előtt az egyes programszegmenseket írjuk le.



Manipulációs szubrutinok

11.41 Az AIR programban felhasznált program szegmensek

Az AIR program a következő rutinokból épül fel.

Az AIR program fő részének program szegmensei:

NUL
OUT1
IN
IANAL
ISTAT
OUT2
IALKP

Manipulációs lehetőségeket megvalósító program szegmensek:

IUTORL
IUTLK
IALK
IPPUK
IUBEH
KOSIKI
INTKI

Alkatrészprogram ellenőrzését ellátó szubrutin:

PRT

A programozást kisegítő szubrutinok

MISTAK

ICURS

IOSZT

NULL

KONV1

Az egyes programszegmensek leírása a következő.

Az AIR program fő részének program szegmensei:

1. NUL szubrutin

Fortran bevezető sora:

SUBROUTINE NUL

Működése:

Működése kettős:

1. Képernyő törlése
2. Új API típusu utasítás összeállítására szolgáló tömbök, illetve változók nullázása.

Felhasznált rutinok: NUL

2. OUT1 szubrutin

Fortran bevezető sora:

SUBROUTINE OUT1

Működése: Csak az 1, 2, 3. típusu állapot-

rekordokat dolgozza fel. IR állapotrekord akciósrészben talált szövegpointerek és cursor címek alapján szövegeket írja ki a display képernyőre. A 4-es típusu állapotrekord befutása esetén 1. számú hibajelzést ad.

Felhasznált rutinok: JAKP, MOVES, MISTAK, ICURS

3. IN szubrutin

Fortran bevezető sora:

SUBROUTINE IN

Működése: Az IR állapotrekordban előírt display helyről beolvassa az operátor által megadott információt és elhelyezi az IPUF tömbbe, a 6., 10. és 62-es altipusu állapotrekordok esetén. Menü típusu állapotrekord esetén a rámutatás helyét, azaz a cursor címét olvassa be.

Hibajelzés: 6. 3. típusu állapotrekordot akar feldolgozni.

4. Állapotrekord altipus száma hibás

5. A beolvasás nem az adott cursor címről történt.

Felhasznált rutinok: JAKP, MISTAK, ICURS, IOSZT

4. IANAL függvény

Neve IANAL

Fortran bevezető sora:

FUNCTION IANAL (IR, IC)

Paraméterei:

IR: egész típusu szám, az aktuális állapotrekord sorszáma

IC: az AIR programban a kapcsoló GOTO utasításra jellemző egész típusu szám

Működése: A beolvasott információkat elemzi, diagnosztizálja, majd az állapotrekordok alapján a függvény értéke megadja az új IR állapotrekord sorszámot és az IC formál paraméter az új IR-hez tartozó új értéket veszi fel.

A 3. típusu állapotrekord esetében, ahol a rendszer nem vár választ az operátortól, a függvény az állapotrekord elemző részében levő IR értéket veszi fel.

A 2. típusu állapotrekord esetében a display-ről beolvasott cursor cím segítségével megkeresi az elemző rész intervallumainak első eleme alapján a válaszhoz tartozó intervallumot. A függvény értéke megadja a megtalált intervallumban lévő IR értéket. A függvény IC formalparamétere pedig az intervallum harmadik elemének értékét veszi fel. Ha a függvény nem találja meg a cursor címnek megfelelő intervallumot, az IANAL az állapotrekordhoz tartozó utolsó intervallumban lévő IR és IC értékeket határozza meg.

Az 1-es típusu állapotrekord esetén a diagnosztika az állapotrekord altipusa szerint történik. 10 és 62 altipusu állapotrekordok esetén a függvény csak azt vizsgálja meg, hogy az állapotrekordba beépített válasz helyről az információ befutott-e vagy sem. Ha a válasz nem futott be, a J változó értékének 1-et, ellenkező esetben 4-et ad. A diagnosztikai vizsgálat után az előzőekben leírtak szerint megkeresi a program a J tartalmához tartozó intervallumot és a függvény az intervallumhoz tartozó IR és IC értéket veszi fel.

A 6-os altipusu állapotrekord esetén
- ha a válasz nem futott be - J változó

1-es értéket kap a fentiek szerint. Ha a válasz befutott, a program megnézi, hogy a befutott információ első karaktere számmal kezdődik-e vagy sem. Ha számmal kezdődik, J változónak 2-es értéket ad, ellenkező esetben folytatja a diagnosztikai vizsgálatot. Egy flage megfelelő állásából megvizsgálja a program, hogy a befutott azonosító egy utasítást identifikál-e, vagy beépül az éppen aktuális APT típusu utasításba. Megnézi, hogy a befutott azonosító szerepelt-e már az azonosítók listáján. Ha a válasz az azonosítók listáján már szerepelt és identifikációs szerepet tölt be, az összeállítandó utasításban J értéke 3-at kap. Ugyancsak 3-at kap J értéke akkor is, ha a válasz az azonosítók listáján nem szerepel és az éppen felépítendő utasításba illeszkedik be. J 4-es értéket kap akkor, ha a befutott válasz a fenti vizsgálatoknak eleget tett, azaz a megadott azonosító hibátlan. A diagnosztikai vizsgálatok után, a program megkeresi a J-hez tartozó intervallumot és abban talált IR és IC értéket veszi fel az IANAL függvény. Minden diagnosztikai vizsgálat előtt a program szegmens megnézi a befutott cursor címet. Amennyiben az a vissza rámutatás cursor címének felel meg, a program a diagnosztikai vizsgálatot nem végzi el, hanem a függvény az állapotrekord VC intervallumnak megfelelő IR és IC értéket veszi fel.

4-es típusu állapotrekord esetén az IANAL megnézi az első intervallum első elemét. Ha 1-est talál, a függvény értéknek a második elem IR értékét adja és IC is felveszi az új értéket. 0 esetén a függvény első lehívásakor az első intervallum második elemének értékét veszi fel és az IC az intervallum harmadik elemének tartalmát kapja meg. A függvény másodszori lehívása esetén, ha a rámutatás nem a vissza sorra volt, a harmadik intervallum első elemében talált IR értéket kapja, a második elemének tartalma kerül IC változóba, míg a harmadik eleme az ITOV változóba.

5. ISTAT szubrutin

Fortran bevezető sora:

SUBROUTINE ISTAT

Működése: A 4. típusu állapotrekord alapján összeállítja a dialógushoz tartozó utasítás részt, amelyet az IUTOMB-tömbben tárol. Az állapotrekordban szereplő szövegpointerekkel jellemzett APT típusu szavakat a szövegtáblából veszi, míg a 9999-el jelzett mutató esetében az operátor által adott beolvasott információt építi be az utasítás részbe. Ha az állapotrekord nem 4-es típusu, a szubrutin hibajelzést ad ki.

Hibajelzések: 8 az állapotrekord típusa
nem 4-es

Felhasznált rutinok: JAKP, IOSZT, MOVES

6. OUT 2 szubrutin

Fortran bevezető sora:

SUBROUTINE OUT2

Működése: Az összeállított kész utasítást a display képernyő utolsó sorába kiírja, felajánlva az operátornak végső ellenőrzésre.

Felhasznált rutinok: -

7. IALKP szubrutin

Fortran bevezető sora:

SUBROUTINE IALKP

Működése: Az operátor által jóváhagyott utasítást egy közös memória mezőbe (IPPR(K)) helyezi el. Ha az összeállítandó utasítás definiáló utasítás volt, az utasítás azonosítóját egy külön memóriában (IAZPUF) tárolja.

Felhasznált rutinok: -

8. PRT szubrutin

Fortran bevezető sora:

SUBROUTINE PRT

Működése: Alkatrészprogram befejezésekor a kész alkatrészprogramról listát, valamint lyukszalagot készít.

Felhasznált rutinok: -

Manipulációs lehetőségeket megvalósító program szegmensek.

9. IUTORL szubrutin

Fortran bevezető sora:

SUBROUTINE IUTORL

Működése: Az operátor által sorszámokkal vagy azonosítókkal megjelölt utasításokat törli az utasítás listáról, és ha azok definiáló utasítások voltak, azono-

sitójukat is az azonosítók listájáról.
Az utasítások sorszámát a megváltozott
viszonyoknak megfelelően átcimzi.

Felhasznált szubrutin: -

10. IUTLK szubrutin

Fortran bevezető sora:

SUBROUTINE IUTLK

Működése: Az összeállított utasítások közül
10 utasítást jelentet meg a display kép-
ernyőn. A megjelentetés az operátor vá-
laszától függően történhet az alkatrész-
program elejétől, végétől kezdve, vagy
tetszőleges sorszámtól kezdve az alkat-
részprogram eleje, vagy vége felé halad-
va. A szubrutin lehetővé teszi a cikli-
kus megjelentetést is, amikor az egyes
ciklusokat az operátor rámutatása gene-
rálja.

Felhasznált szubrutin: -

11. IALK szubrutin

Fortran bevezető sora:

SUBROUTINE IALK

Működése: A szubrutin hívásáig összeállított
alkatrészprogram utasításokban szereplő
azonosítókat jelenteti meg a display kép-
ernyőn, az őket definiáló utasítások sor-
számával együtt.

Felhasznált szubrutinok: MOVES

12. IPPUK szubrutin

Fortran változat kezdő sora:

SUBROUTINE IPPUK

Működése: Akár azonosítóval, akár sorszá-
mal megjelölt alkatrészprogram utasítást
jelentet meg a display képernyőn.

Felhasznált szubrutinok: MOVES, KONV 1.

13. IUBEH szubrutin

Fortran változat kezdő sora:

SUBROUTINE IUBEH

Működése: A szubrutin az addig megépített alkatrészprogram utolsó egy vagy több utasítását helyezi el az operátor által sorszámmal megadott helyre. Az utasítások áthelyezéséből származó sorszámmódosításokat az alkatrészprogram listán, valamint az azonosítók listáján a szubrutin elvégzi.

14. KOSIKI szubrutin

Fortran változat kezdő sora:

SUBROUTINE KOSIKI

Működése: Az operátor válaszáától függően az alkatrészprogram utasításai közül az utoljára talált koordinátarendszer transzformációs utasítást vagy a ZSURF utasítást írja ki a display képernyőre, amelyből az operátor tájékoztatást kaphat arra, hogy mely koordinátarendszerben, illetve síkban dolgozik.

Felhasznált program szegmensek: NULL MOVES
IOSZT

15. INTKI szubrutin

Fortran változat első sora:

SUBROUTINE INTKI

Működése: A képernyő adott sorába beirt APT típusu utasítást ellenőrzés nélkül bemásolja az utasítás listára, miközben az utasítószámlálót eggyel megnöveli.

Felhasznált program szegmensek: NULL

Kisegítő program szegmensek.

16. MISTAK szubrutin

Fortran változat első sora:

SUBROUTINE MISTAK (I, K)

Működése: Az állapotrekordok ellenőrzése közben fellépő hibák kiiratására szolgáló szubrutin. Formál paraméterei közül az I=0 informatív hiba kiírását végzi és a programot hagyja tovább futni. I=1 esetén a hiba fatális, a program futását letiltja. K formál paraméter a hiba szöveg kódszámát adja meg.

17. ICURS szubrutin

Fortran változat első sora:

SUBROUTIN ICURS

Működése: A display képernyőről beolvasott cursor címből meghatározza a képernyő sor, illetve oszlop számot.

18. IOSZT függvény

Fortran változat első sora:

FUNCTION IOSZT (I)

Működése: Jelentősége csak a szimulációs programban van. J a szimulált képernyő azon kezdő oszlopcímet adja meg, amely I-hez tartozik úgy, hogy a kiírandó szövegek egymást ne írják felül.

19. NULL szubrutin

Fortran változat kezdő sora:

SUBROUTINE NULL

Működése: a képernyő nullázása.

20. KONV 1. szubrutin

Fortran változat kezdő sora:

SUBROUTINE KONV 1. (IPUF)

Működése: Jelentősége csak a szimulációs programban van. Az IPUF tömbbe "A" formátummal beolvasott szám információt integer számmá alakítja át.

A programok FORTRAN listáját a 4. függelék tartalmazza.

11.42 Az AIR program működése

Az AIR program működését a 24. ábra alapján mutatjuk be. A kezdő értékek beállítása után a NUL szubrutin egyrészt nullázza a display képernyőt, valamint az induló dialógushoz szükséges változók értékeit állítja be. Az aktuális IR sorszámu állapotrekord alapján az OUT1 szubrutin kiírja a képernyőre a dialógushoz tartozó kérdést. Az IN szubrutin beolvassa az operátor válaszát, amelyet az IANAL függvény diagnosztizál, meghatározza az új IR állapotrekord sorszámot és a hozzá tartozó IC kapcsoló GOTO-ra jellemző számot. Ha a feltett kérdés egy újabb kérdéssel folytatódik, a program a 2-es címkére ugrik. Ha az operátor a képernyő VISSZA sorára mutat rá, vagyis a dialógusba beépített visszacsatolásra, az 1-es címkén folytatódik a program futása. Hibás vagy hibátlan válasz esetén a 7-es címkén keresztül jut el a program az OUT1 szubrutinra, amely a 3-as típusu állapotrekord alapján vagy a hibakiírást végzi el, vagy a képernyő megfelelő sorait törli. Hibakiírás esetén a program futása az OUT1 szubrutinon keresztül a 3-as

cimkén folytatódik, míg hibátlan válasz esetén a 4-es cimkén. OUT1 szubrutin kihagyásával a program futása a 4-es cimkén folytatódhat akkor, ha sem hibakiírásra, sem képernyő törlésre nincs szükség. A részdialógushoz tartozó APT utasítást az ISTAT szubrutin állítja össze a 4-es típusú állapotrekord szerint. Ha az utasítás felépítéséhez még további dialógusok tartoznak, az IANAL függvényen keresztül a program futása a 2-es cimkén folytatódik. Ha az utasítás összeállítására több aldialógusra már nincs szükség, azaz az utasítás felépítése befejeződött, az 5-ös cimkére ugrik a program. A részdialógusok alatt összeállt és a KONVS szubrutin által összevont utasítást az OUT2 szubrutin kiírja a display képernyőre ellenőrzés céljára. A kiírt utasítást az operátor vagy elfogadja, vagy nem. Válaszát az IN szubrutin olvassa be. Ha az operátor nem fogadja el a kiírt utasítást és a display VISSZA sorára mutat, a program futása az 1-es cimkén folytatódik, törölve az összeállított és kiírt utasítást. Ha a rámutatás nem a VISSZA sorra történt, a rendszer azaz az IANAL függvény elfogadottnak tekinti az utasítást és a program futása a 6-os cimkén folytatódik.

IALKP szubrutin az utasítás listába elhelyezi az operátor által elfogadott utasítást. Az alkatrészprogram írása attól függően, hogy az befejeződött-e vagy sem, az 1-es cimkén vagy a 8-as cimkén folytatódik. A befejezett alkatrészprogram a PRT szubrutinban még egy utolsó ellenőrzésre kerül. Hibátlan alkatrészprogramról listát, valamint lyukszalagot készít, amivel a dialógus befejeződik.

Manipulációs szubrutinok aktivizálása a manipulációs dialógusokat leíró állapotrekordokon keresztül történik.

AIR program listája a 4-es függelékben található meg.

12. ÖSSZEFOGLALÁS

Az APT típusu alkatrészprogramíró rendszer két fő részből áll. Az első a TRANSAIR, a DISTAR-B és az AIR rendszerek, a másik a TRANSAIR és a DISTAR-B rendszerek által előállított állapotrekordok és szövegtáblák. Minden olyan APT típusu nyelvre, amelyre ki akarjuk terjeszteni az interaktív alkatrészprogramíró rendszert, külön-külön el kell készíteni az állapotrekordokat és szövegtáblákat.

Egy konkrét APT típusu nyelvre alkalmazva az interaktív alkatrészprogramíró rendszert, a következő munkákat kell elvégezni a dolgozatban leírt irányelvek betartásával.

- A 6. ábrán bemutatott gráf segítségével az adott APT típusu nyelvben talált utasításokat csoportosítani kell (a fa struktúra kialakítása)
- A gráf csomópontjaihoz tartozó menüket, valamint az egyes utasításokat leíró dialógusokat kell megtervezni (lásd 2. függelék). Dialógusképek elkészítése.
- A dialógusképek alapján a dialógusokat kell leírni a TRANSAIR nyelv segítségével (lásd 3. függelék).

Az értekezésben ismertetett elvek és feltételek betartásával három APT típusu nyelvre az EXAPT1-re, a 2CL-re és az ADAPT-re készítettük el az alkatrészprogramíró rendszert. Az állapotrekordokat szegmensekre bontva állítottuk össze. Egy nyelvhez tartozó állapotrekord szegmensek száma 10-14 között van. Egy szegmens körülbelül 150 állapotrekordot és 50 szövegrekordot tartalmaz.

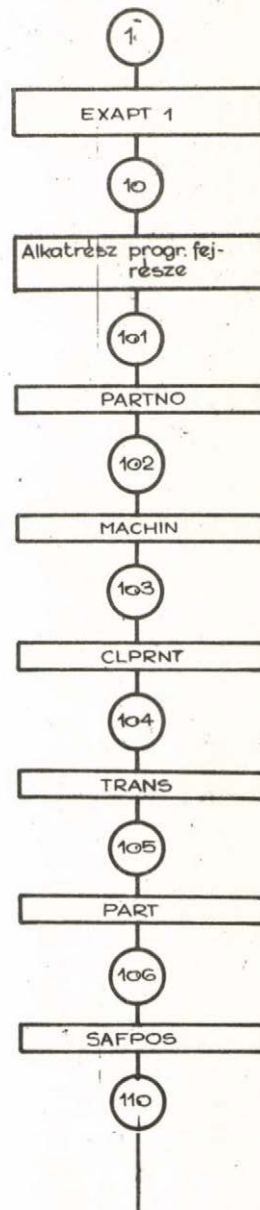
Tapasztalatok szerint egy közepes bonyolultságú alkatrészprogram 15-20 perc alatt elkészíthető a rendszer segítségével, amely már szintaktikusan teljesen hibátlan, szemantikus hibáktól pedig nagyrészt mentes.

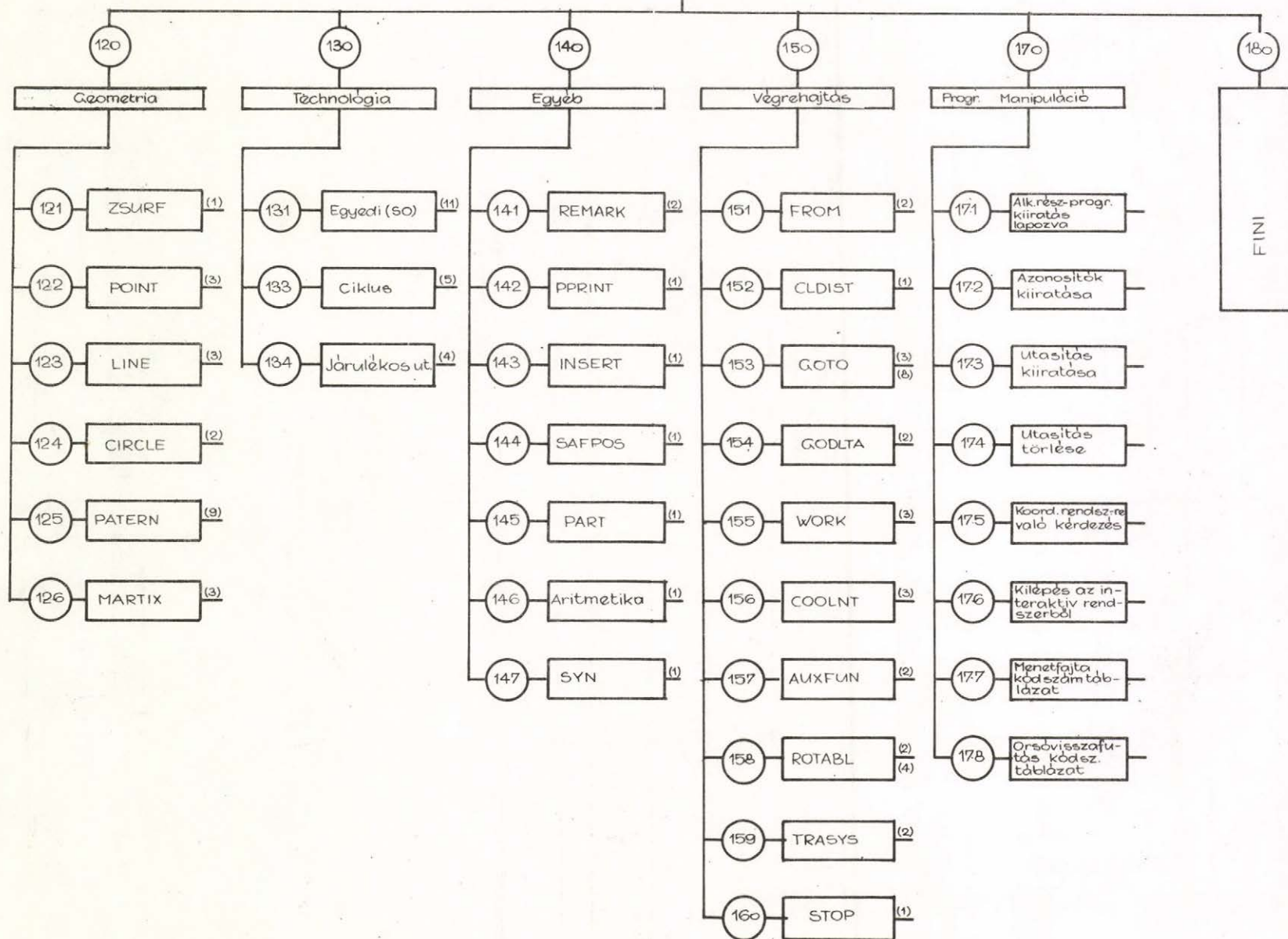
1. Függelék

AZ EXAPT1 NYELV UTASITÁSKÉSZLETÉNEK
FELOSZTÁSA

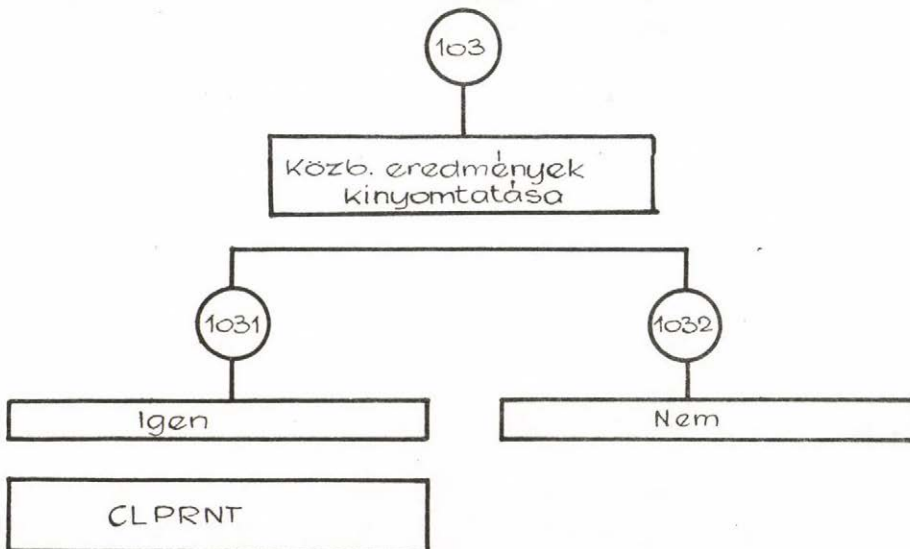
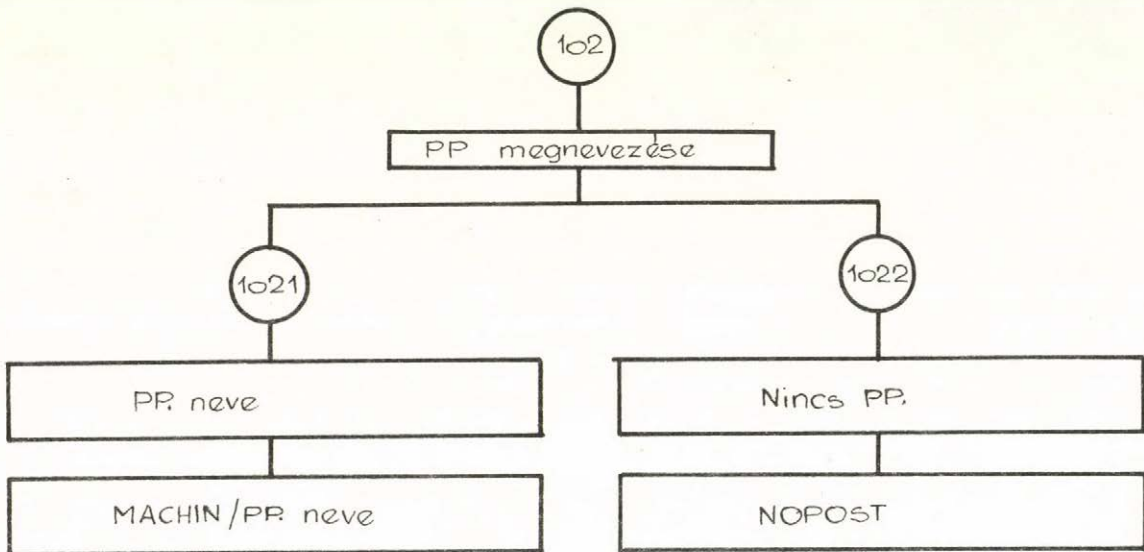
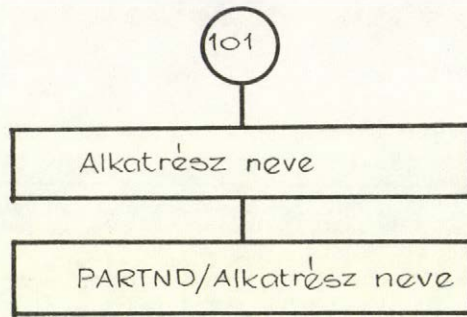
Az 1. függelékben bemutatjuk az EXAPT1 nyelv utasításkészletének felosztását, amelyet a dolgozat 10.3 fejezetében ismertetett elvek alapján készítettünk el. A gráf fejrészában található azok az utasítások, amelyek megadása az alkatrész-programban kötelezőek. Az 1. ábra 110-es csomópontjában helyezkedik el az első menü. Az első menü tagjai a 120, 130, 140, 150, 170 és a 180 számokkal jellemzett csomópontok. E számokkal jellemzett csomópontokhoz szintén menüket rendelünk, amelyek a fa struktúra második szintjét alkotják. A második szintű menük nagy részéhez még egy harmadik szintű menüt rendelünk. A függelék 1. ábrájában szereplő zárójeles számok mutatják meg, hogy az egyes csomópontokba hány újabb faág fut össze.

Az utasításkészlet teljes felosztását a függelék 1-22. ábrája tartalmazza, amelyek közti összefüggést az 1. ábrán bevezetett számrendszer teremti meg.

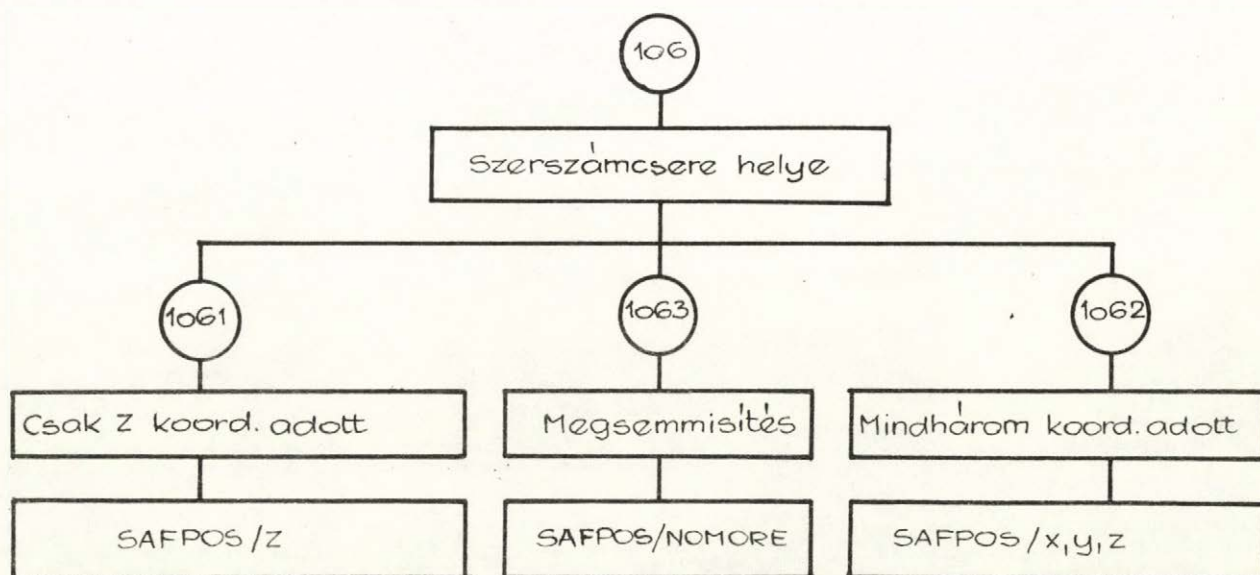
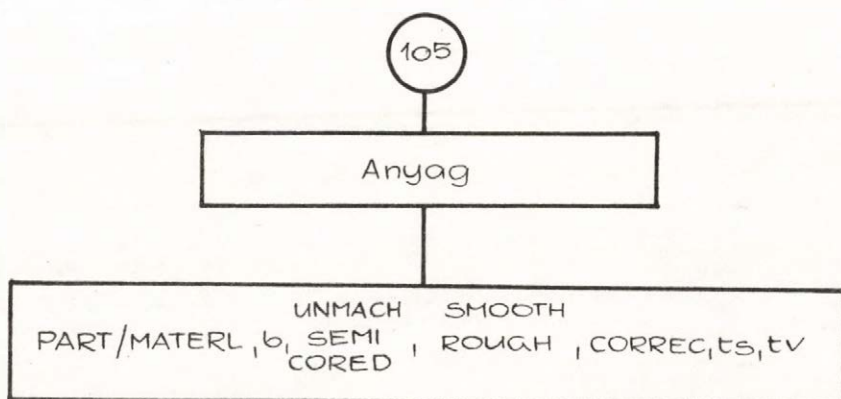
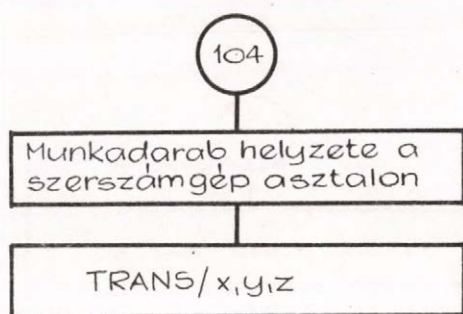




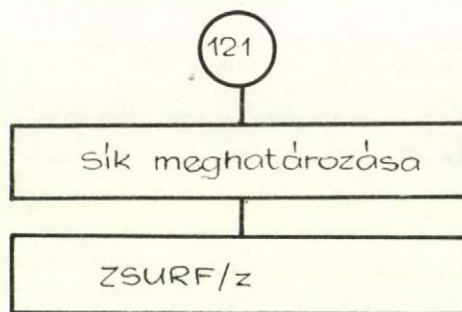
1. ábra



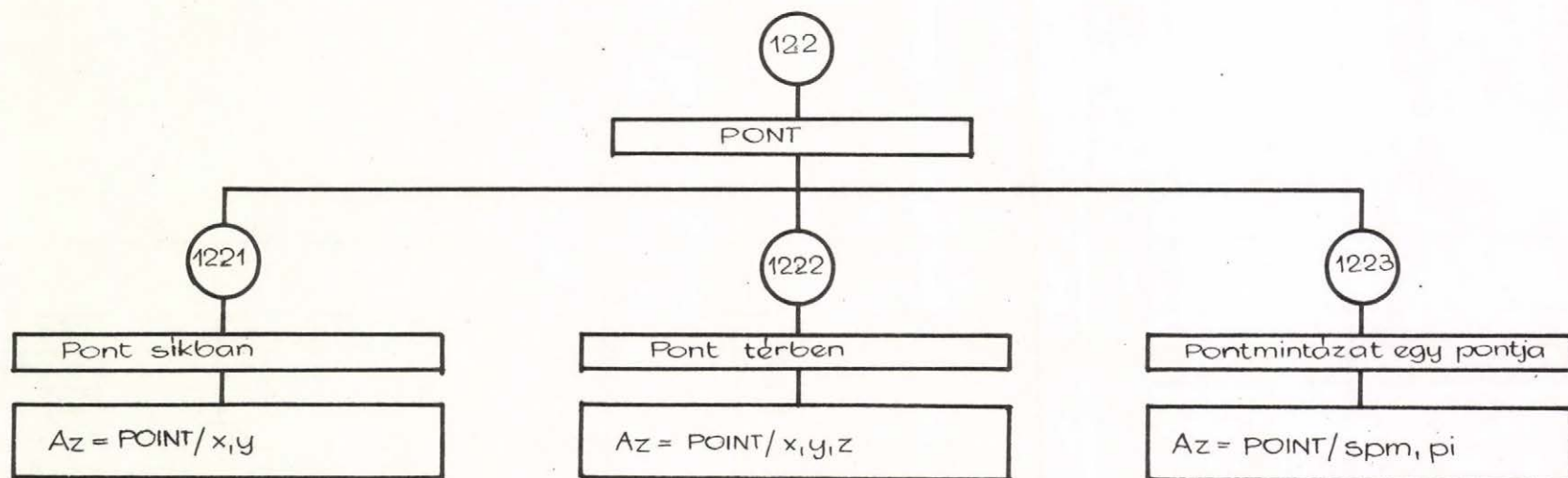
2. ábra



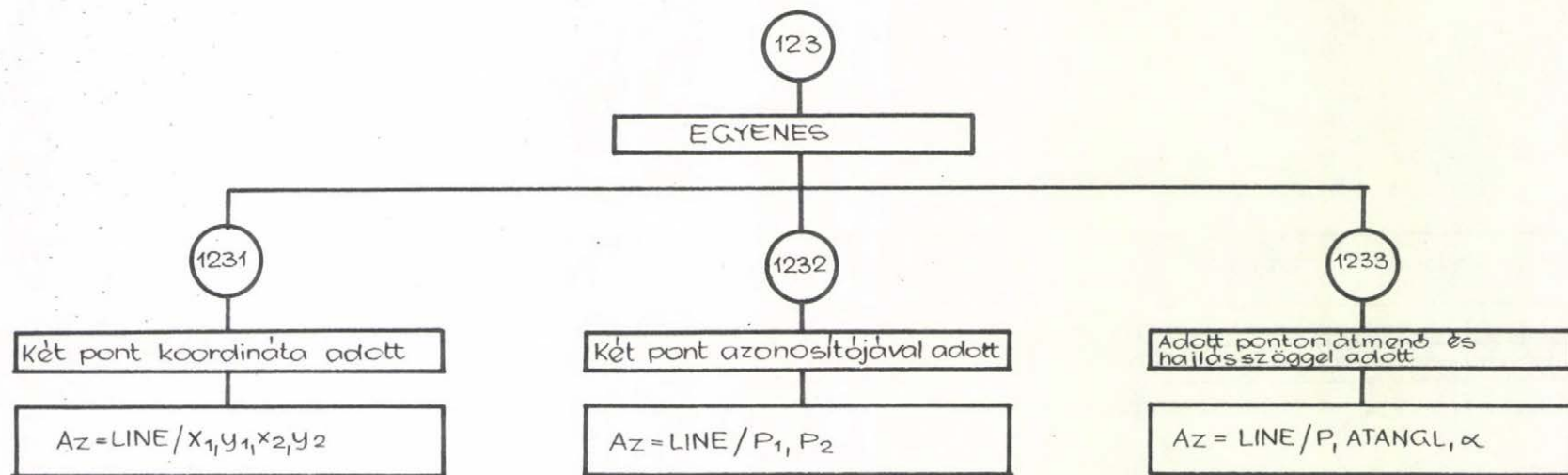
3. ábra



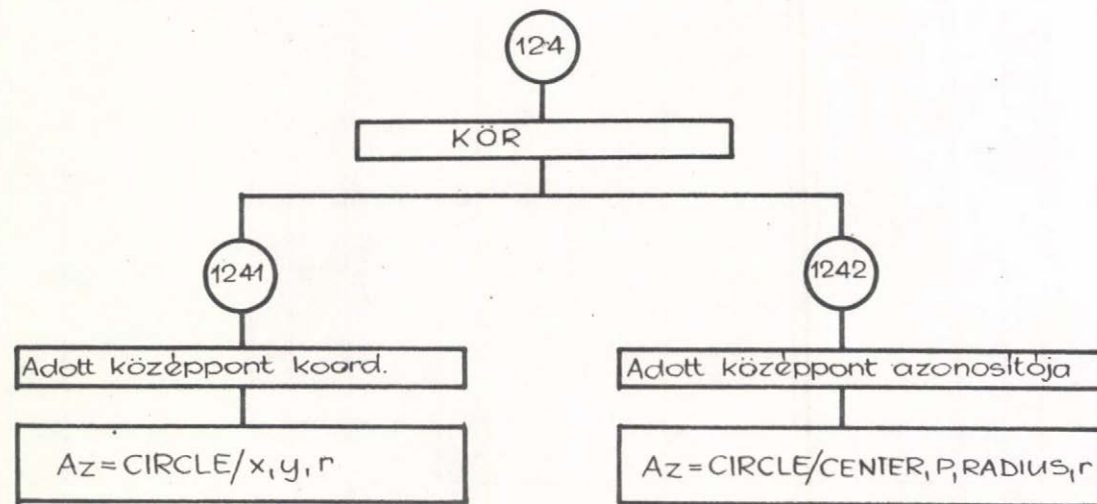
4. ábra



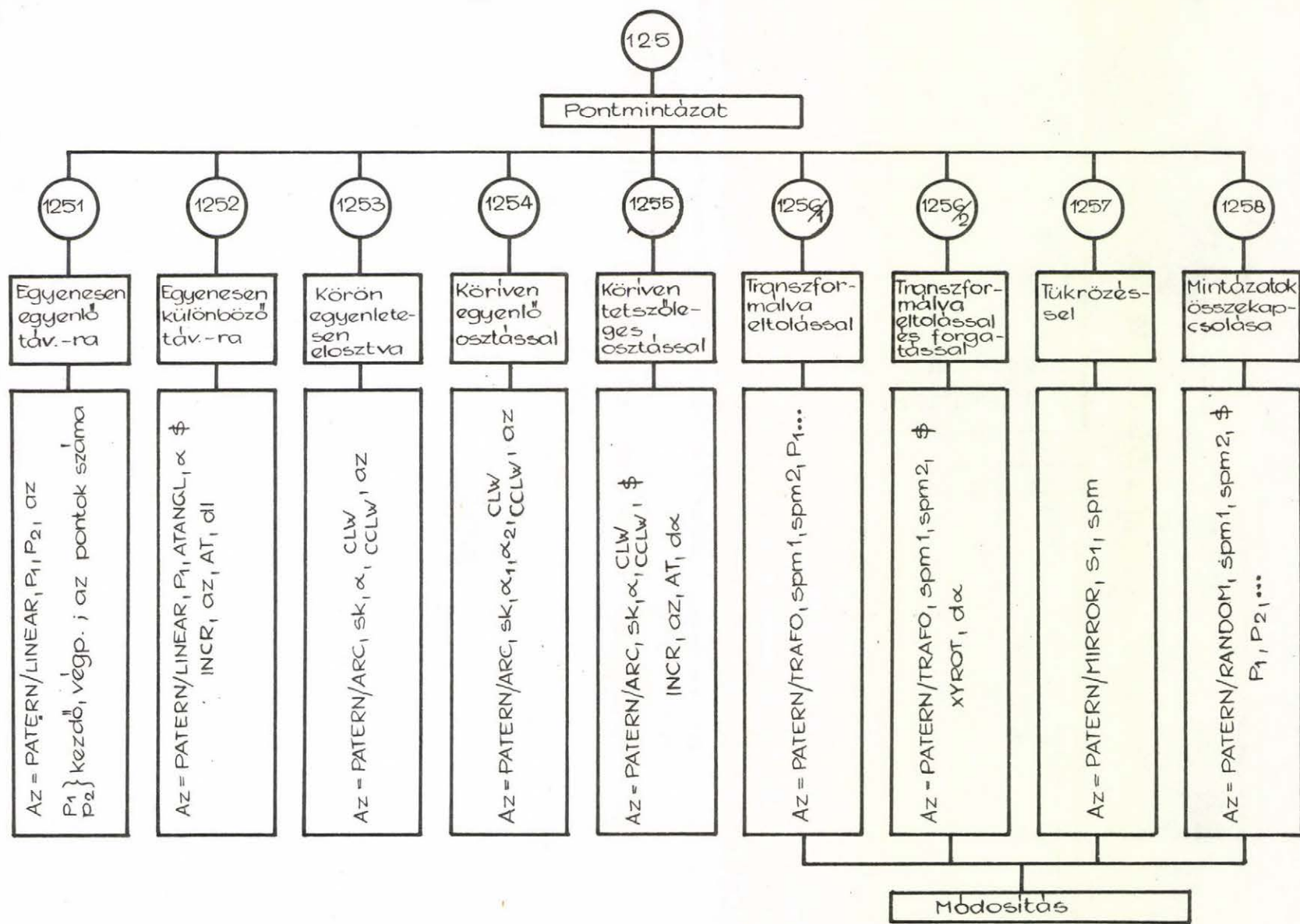
5. ábra



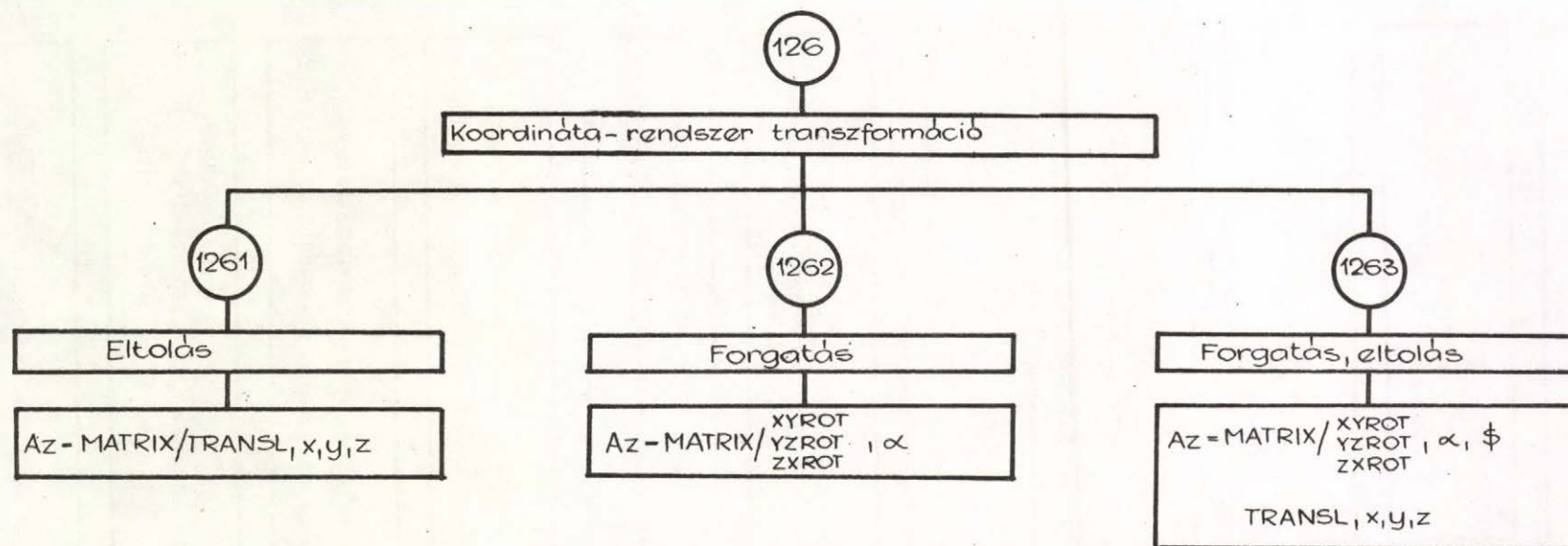
6. ábra



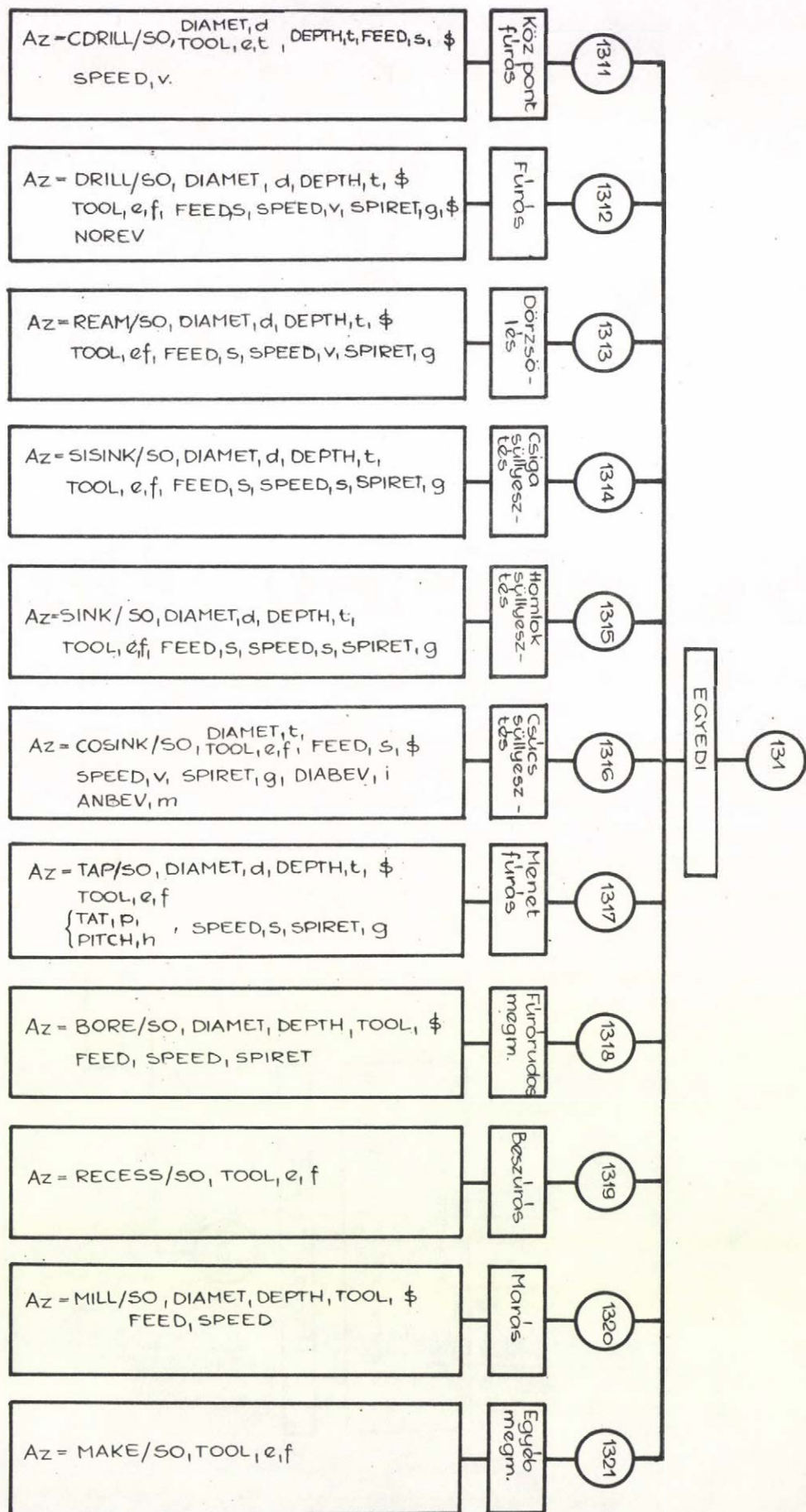
7. ábra

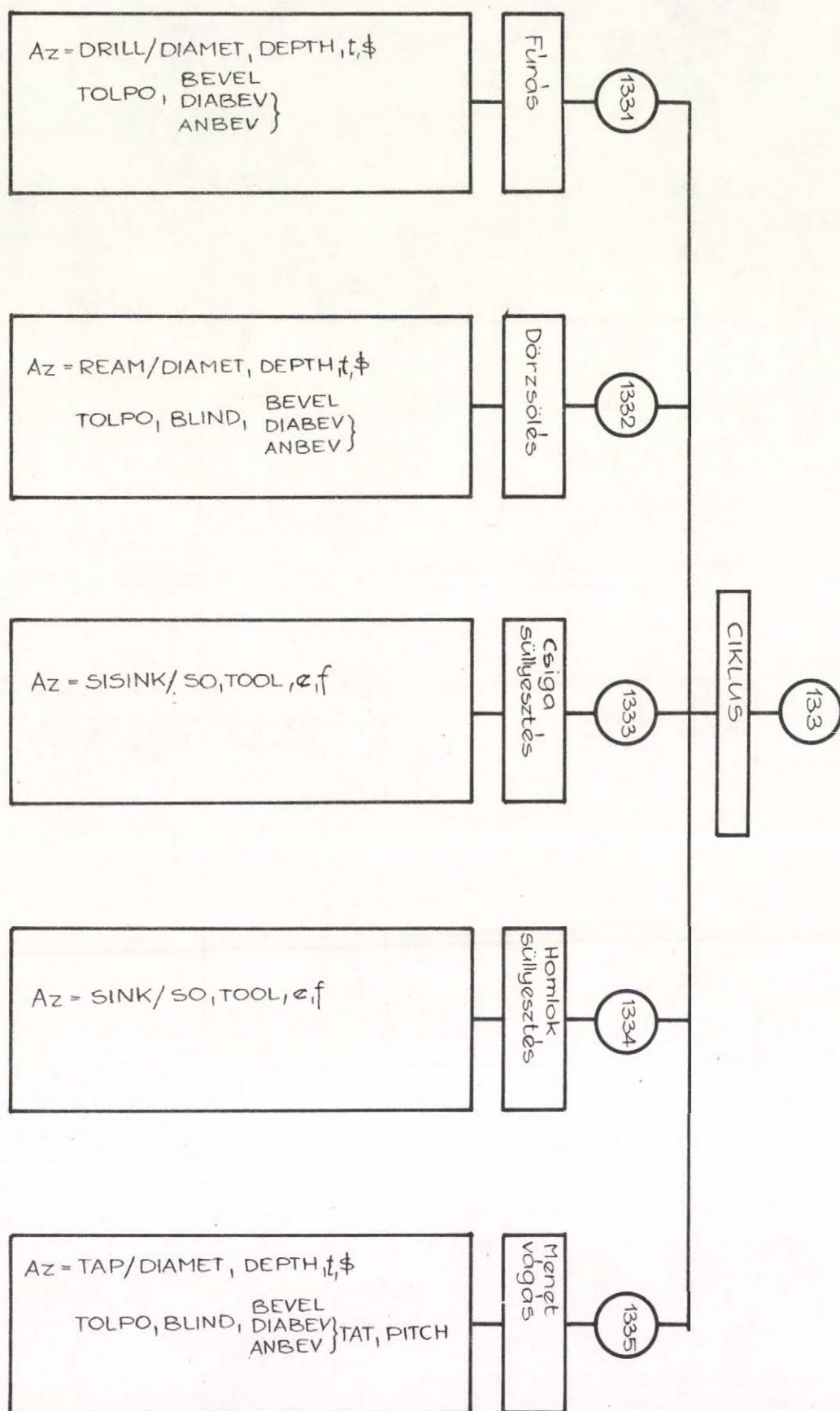


8. ábra

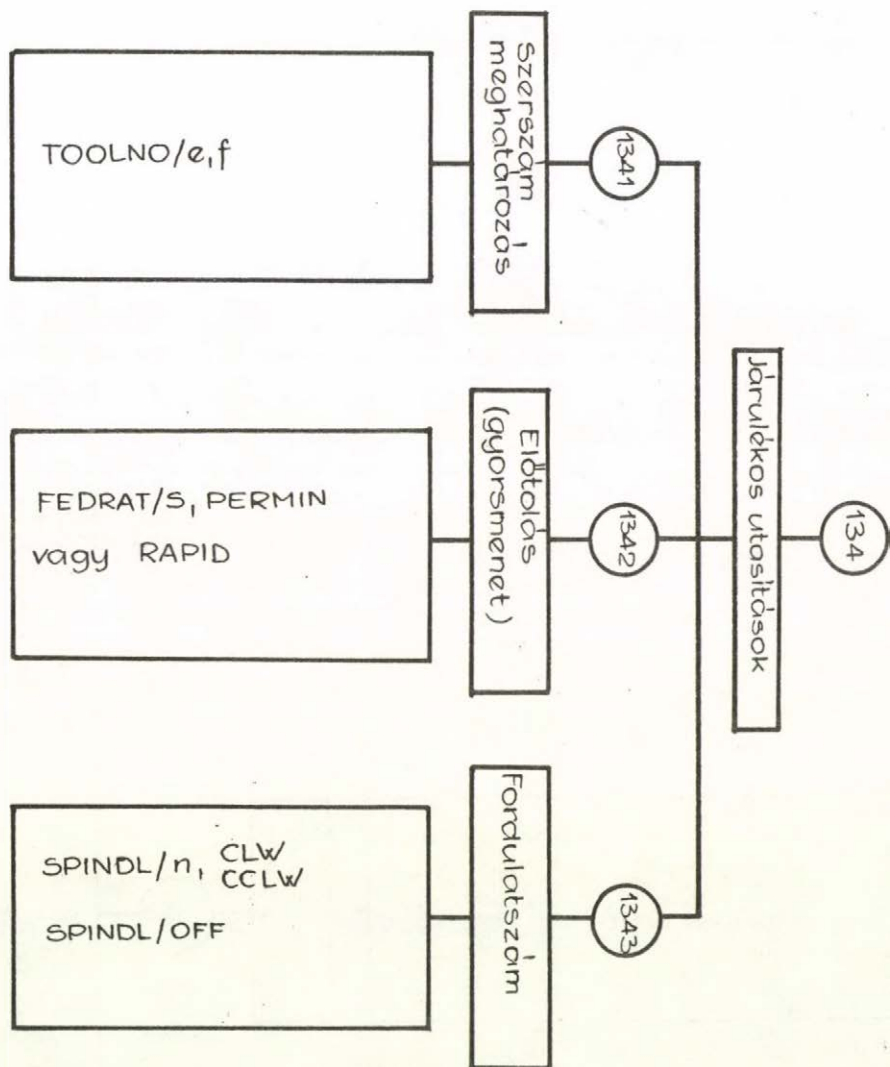


9. ábra

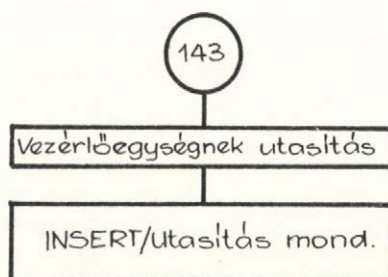
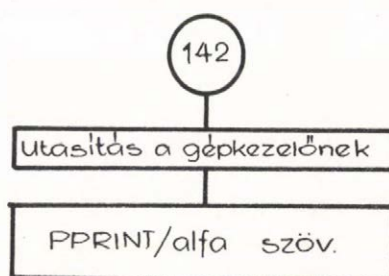
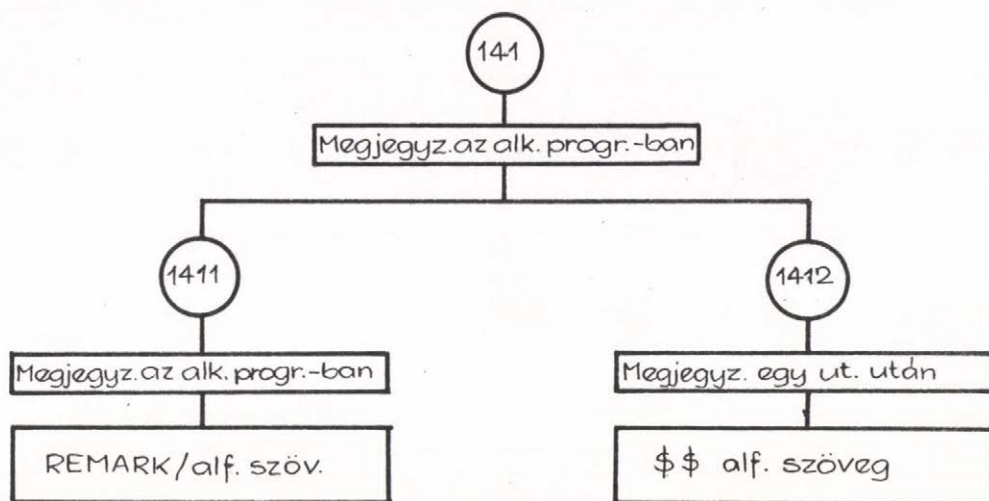




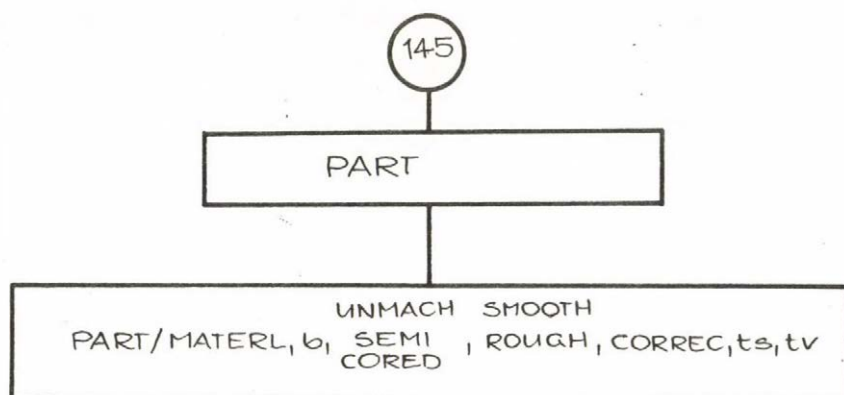
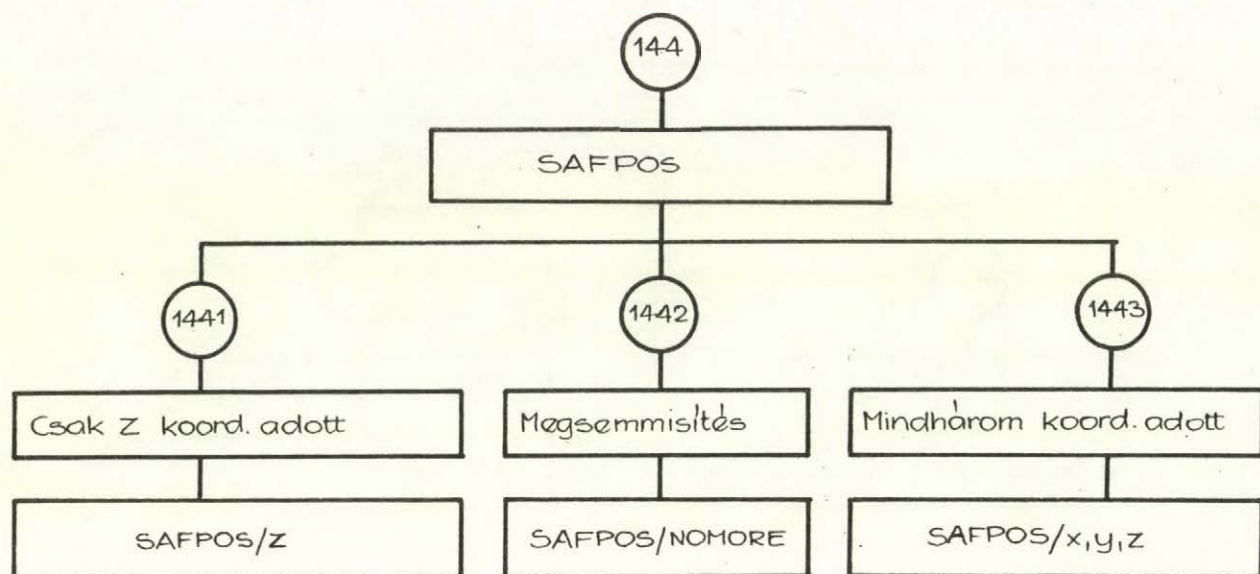
11. ábra



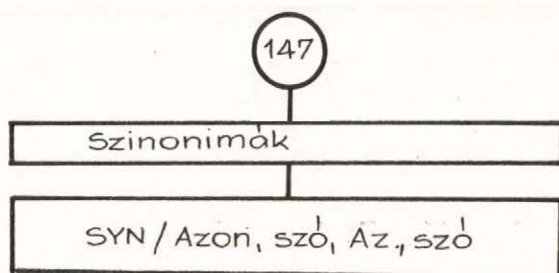
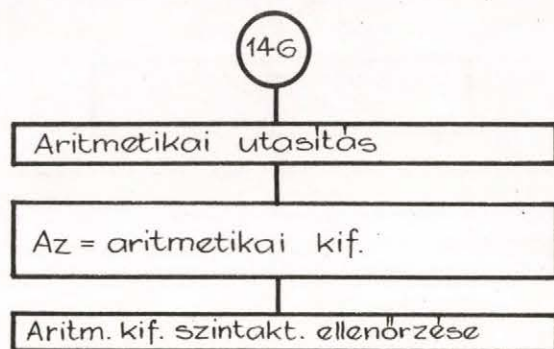
12. ábra



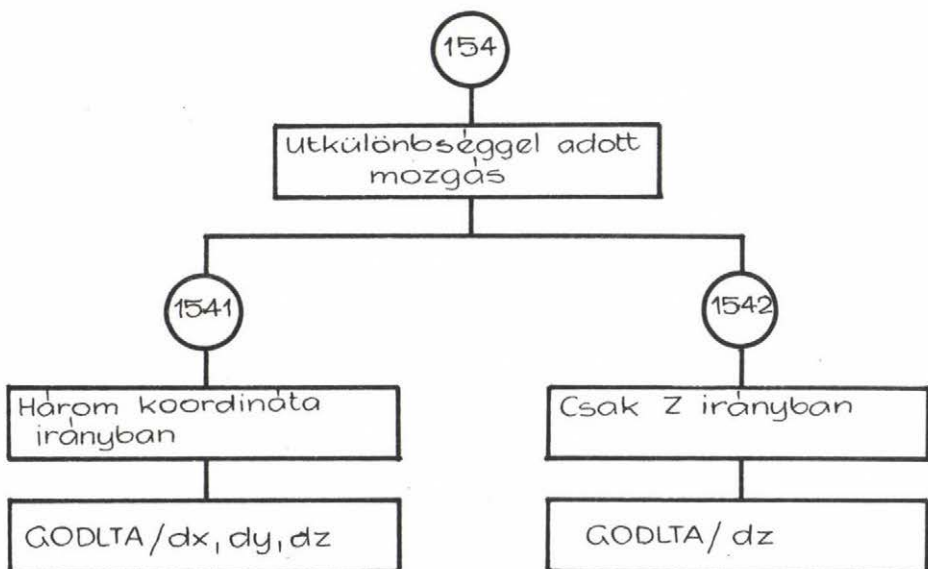
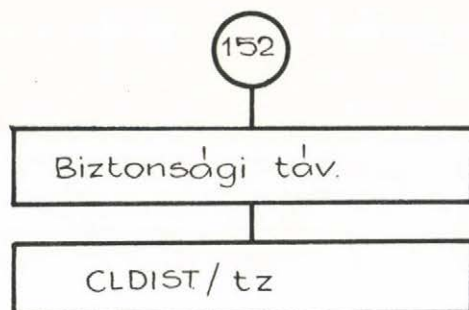
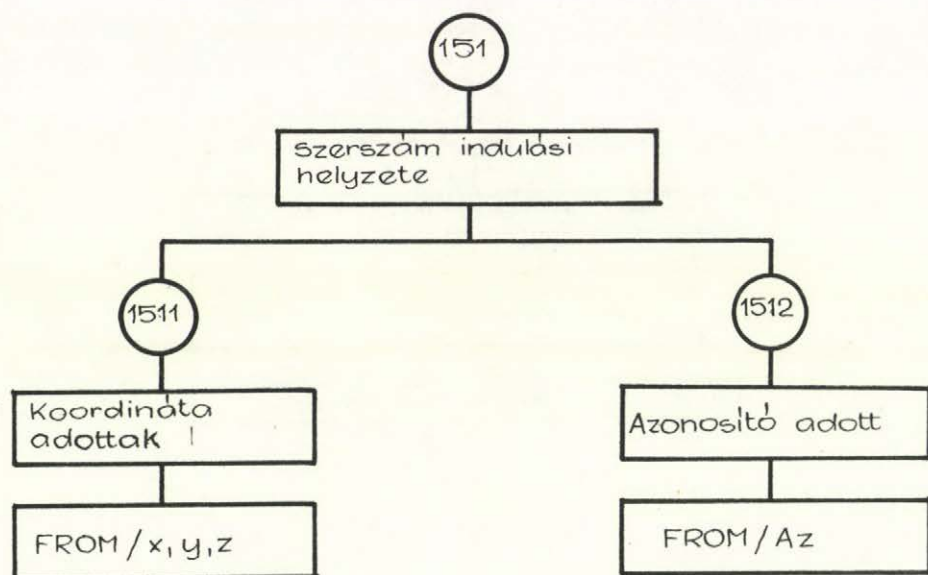
13. ábra



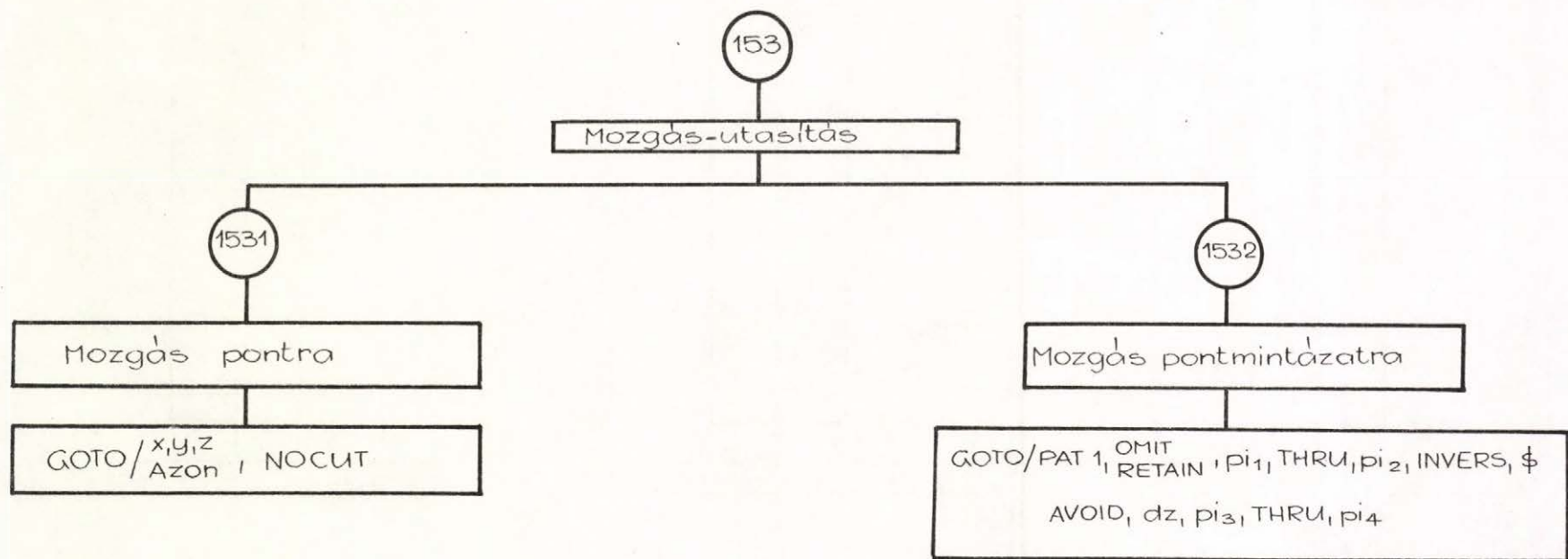
14. ábra



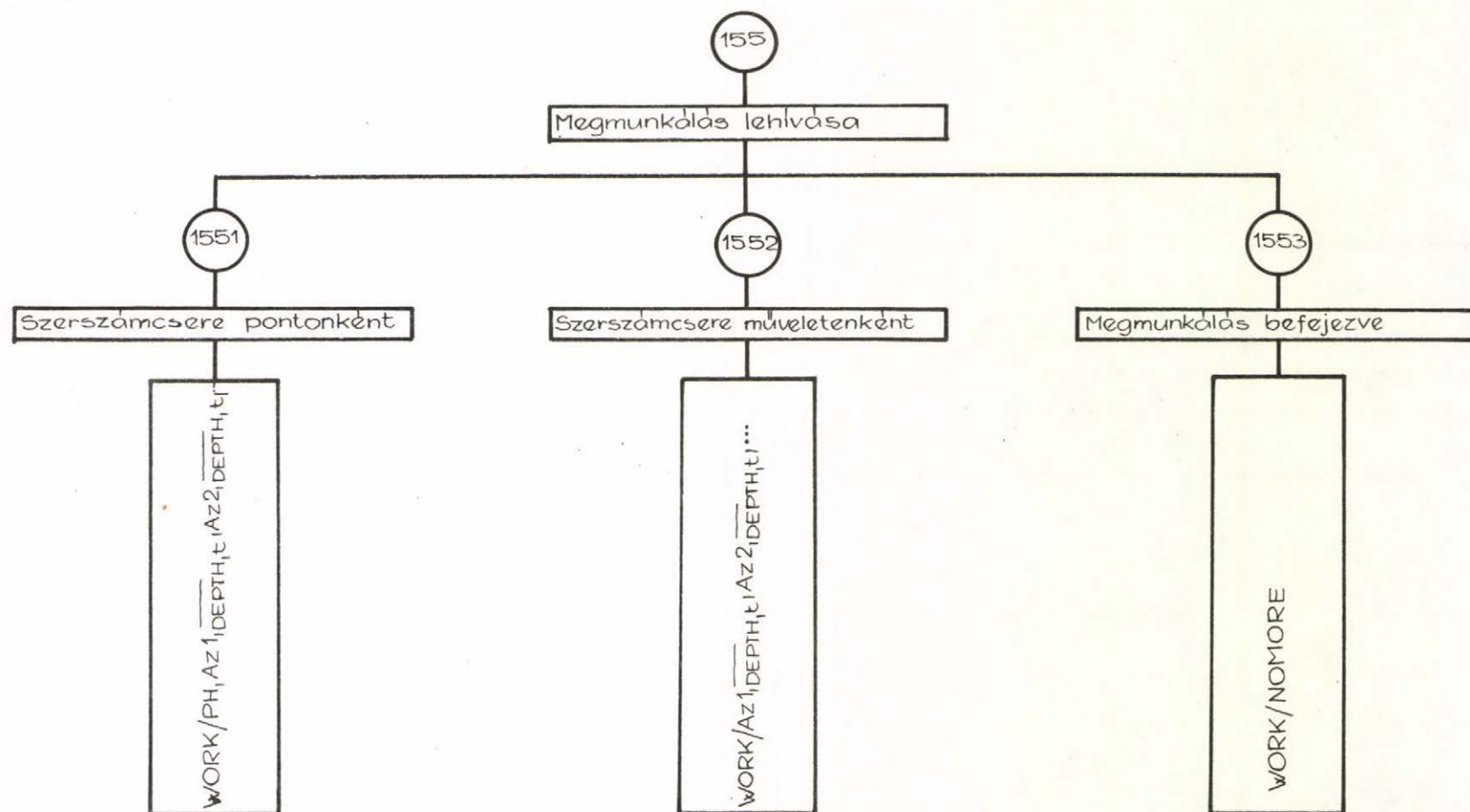
15. ábra



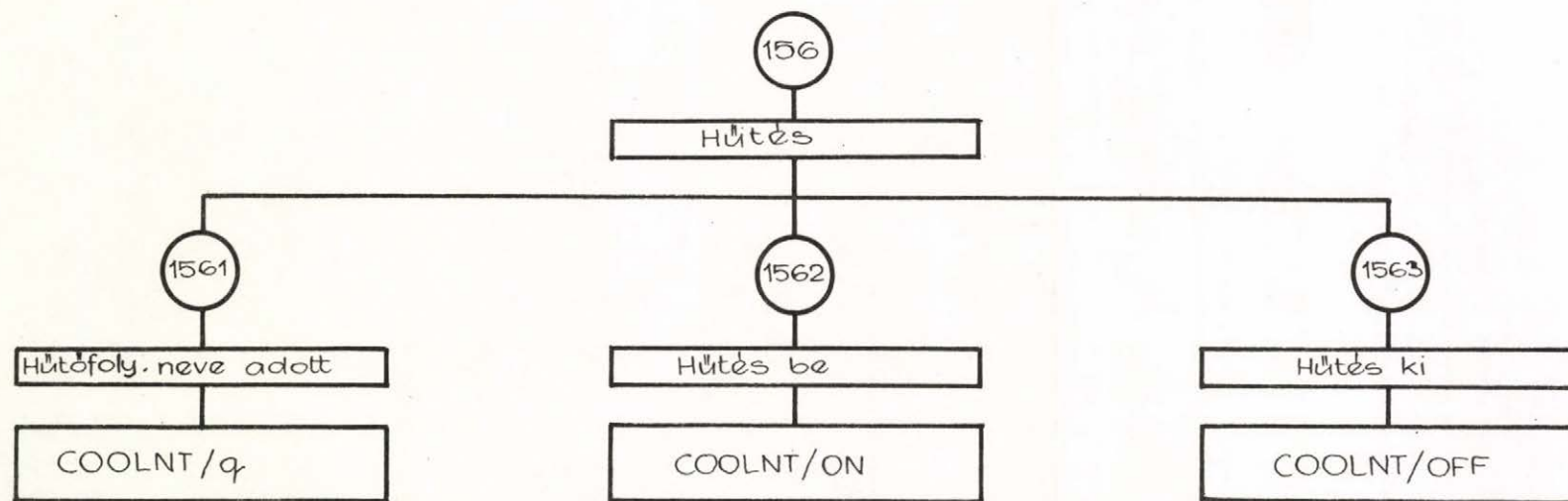
16. ábra



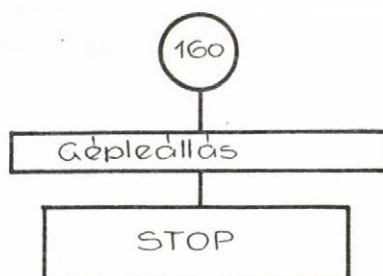
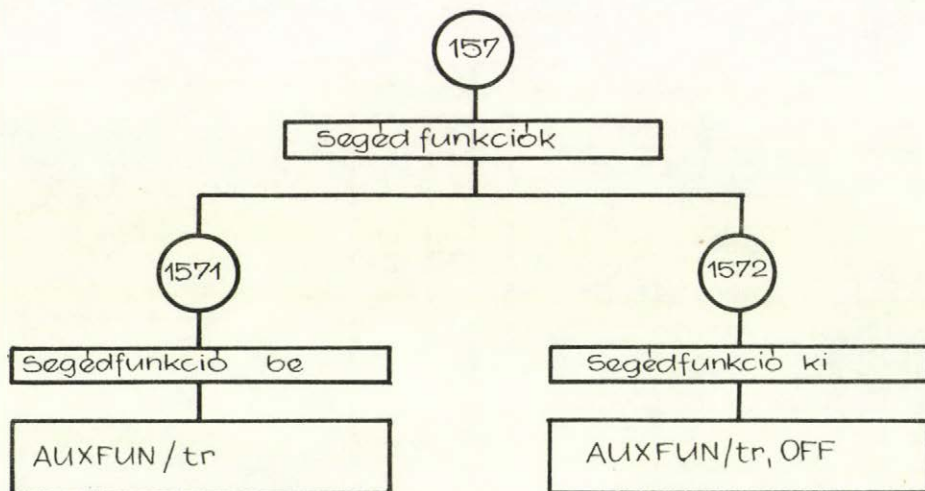
17. ábra



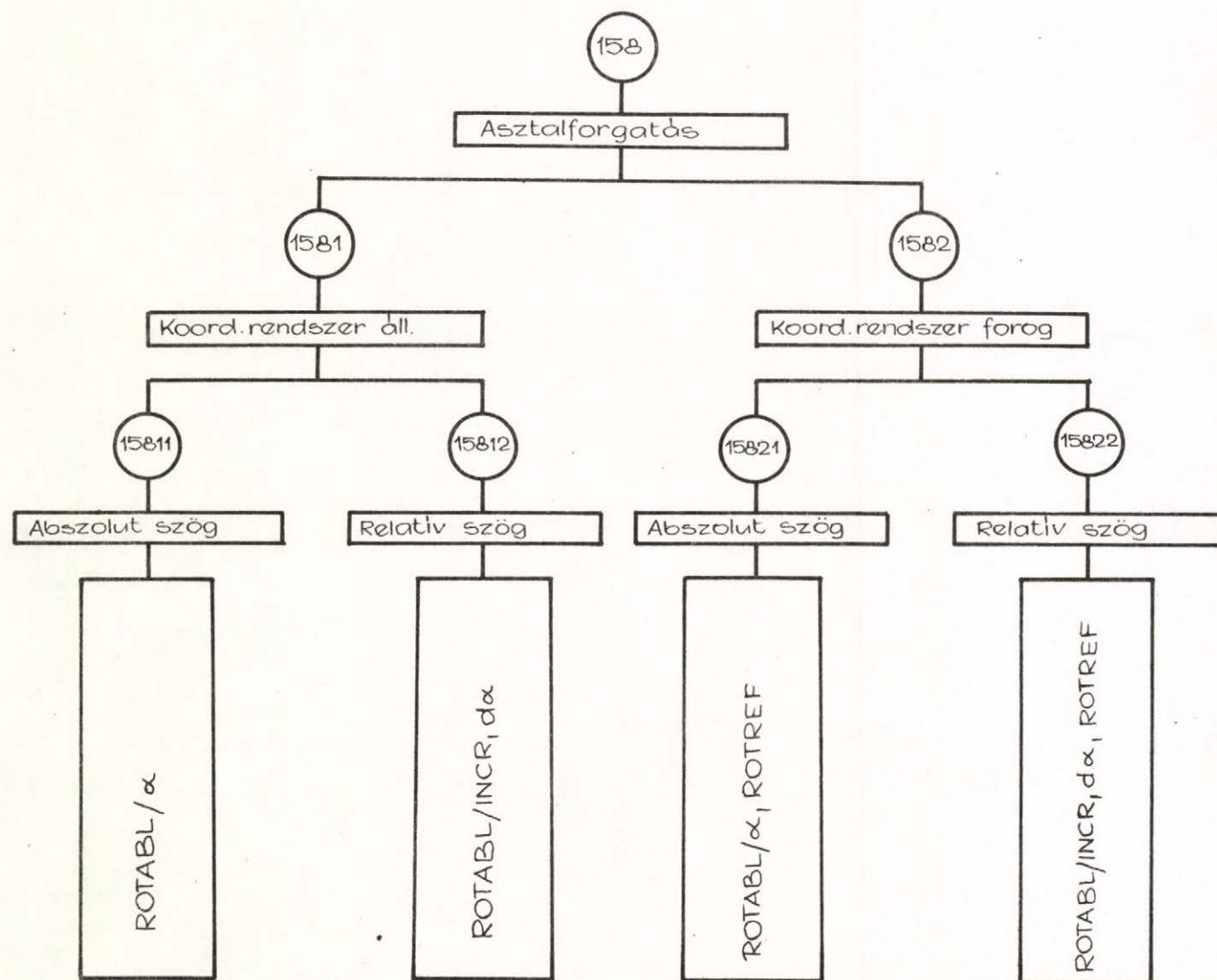
18. ábra

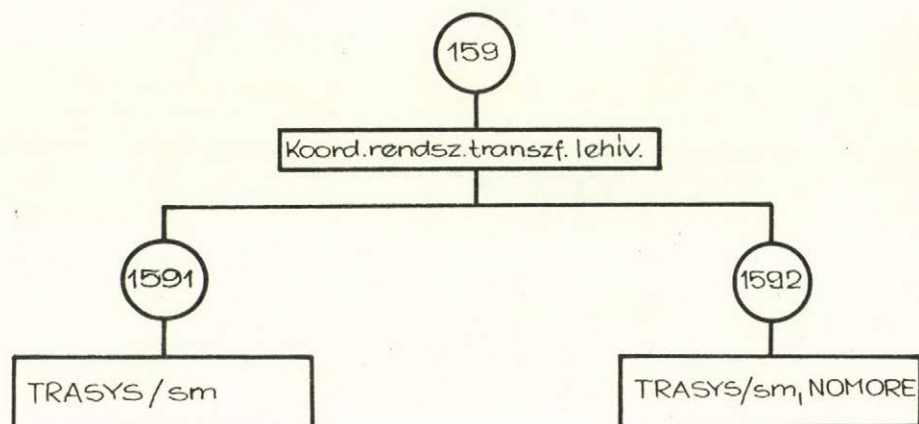


19. ábra



20. ábra





22. ábra

2. Függelék

NÉHÁNY EXAPT1 UTASÍTÁS MEGHATÁROZÁSÁHOZ TERVEZETT
DISPLAY DIALÓGUSKÉPEK

A képernyő képek kialakításának szempontjait 10.4 fejezetben foglaltuk össze. Az ott leírt szempontok alapján mutatunk be néhány EXAPT1 utasítás megkeresésére és felépítésére szolgáló képernyő tervet, amely egyszerű dialógusokból, vagy menükből áll.

E képek a dialógusok leírásának alapjait képezik.

PARINO

AZ ALKAITRESZ NEVE:

<

>

VISSZA < >

MACHIN

POST-PROCESSOR

VAN

< >

N | N | C | S

 \angle \angle

POST-PROCESSOR NEVE:

Λ

Δ

V I S S Z A < >

CLIPRNT

CLTAPE-ROL LISTAT KEREK

< >

NEM KEREK

< >

VISSZA < >

TRANS

ALKATRÉSZ KOORDINÁTERENDSZER ORIGÓJA

A SZÍRSZÁMGEPI ASZTALON

X: < >

Y: < >

Z: < >

VISSZA < >

		PART
ALKATRÉSZ ANYAGÁNAK SZÁMA		< >
MEGMUNKÁLÁSI HELYEK	1. ELOMUNKALATLANOK	< >
	2. ELOMUNKALTAK	< >
	3. ELOONTOTTEK	< >
MUNKADIÁRIÁB FELÜLETE	1. MEGMUNKALT	< >
	2. NYERS	< >
FORGÁCSOLÁSI PARAMÉTEREK MODOSÍTÁSA:		
	1. ELOTOLÁS	< >
	2. FORDULATSZÁM	< >
A KÉRDÉSRE NEM KÖTELEZŐ VÁLASZOLNI!	VISSZA	< >

SAFPOS

SZERSZÁMCSERE HELYE

EGY PONTBAN

< >

SÍKBAN

< >

PONT KOORDINATAI:

X:

<

>

Y:

<

>

Z:

<

>

VISSZA

< >

SAFPOS

SZERSZAIMCSERE HELYE

EGY PONTBAN

< >

SIKBAN

< >

SIK Z KOORDINATAJA

< >

VISSZA < >

MELYIK FEJEZETET KERED?

1. GEOMETRIAI MEGHATÁROZÁSOK

< >

2. TECHNOLÓGIAI MEGHATÁROZÁSOK

< >

3. LEHETŐSÉGEK, VEGREHÁZÍTÁSOK

< >

4. EGYEB PROGRAMTECHNIKAI UTASÍTÁSOK

< >

5. PROGRAMMANIPULÁCIÓK

< >

6. ALKATRÉSZEK PROGRAM VEGE

< >

VISSZA < >

MELYIK FEJEZETET KERED?

MEGHATÁROZÁSOK:

1. SÍK

< >

2. EGYEDI PONT

< >

3. EGYENES

< >

4. KOR

< >

5. PONTMINTAZAT

< >

6. KOORDINÁTERENDSZER TRANSZFORMÁCIÓ

< >

IV. SZÁM

< >

POINT

A PONT ELHELYEZKEDIK:

1. SÍKBAN

< >

2. TERBEN

< >

3. PONTMINTÁZAIT EGY ELEMEN

< >

VISSZA

< >

										POINT																			
A PONT										AZONOSÍTÓJA:										< >									
										KOORDINATAI:																			
																				X: < >									
																				Y: < >									
																				Z: < >									
																				VISSZA < >									

VALASZOLE!

1. UJABB PONTOT AKAROK MEGHATAROZNI:

< >

2. MAS GEOMETRIAI FEJEZETET KEREM:

< >

3. GEOMETRIAI RESZDOL KILEPEK:

< >

VISSZA < >

3. Függelék

TRANSAIR valamint az AIR közbelső nyelveken írt
dialógusprogramok

A 3. függelékben mutatjuk be a TRANSAIR loader program inputját, illetve az outputját. A TRANSAIR dialógus nyelv a loader program inputja, az AIR közbelső dialógus nyelv a loader program outputja.

A TRANSAIR dialógus nyelvet a dolgozat 11.33 fejezetében, az AIR közbelső dialógus nyelvet a 11.2 fejezetben ismertettük. A közölt programrészek a 2. fejezetben bemutatott display dialógusképeket írják le.

P EXAPT
1Q 058PARTNO
Q ALKATRESZ NEVE:
A 562
R PARTNO/
2Q 058MACHIN
Q 300POST-PROCESSOR
Q 350VAN
Q 450NINCS
J 3 *3
J 4 *4
3Q 058MACHIN
Q 300POST-PROCESSOR
Q 350VAN <↑>
Q 350NINCS < >
Q 500POST-PROCESSOR NEVE
A 662
R MACHIN/
G *5
4Q 058MACHIN
Q 300POST-PROCESSOR
Q 350VAN < >
Q 350NINCS <↑>
M99 NOPOST
5Q 058CLPRNT
Q 300CLTAPE-ROL LISTAT KEREK
Q 414NEM KEREK
M 3 CLPRNT
MOO
6Q 058TRANS
Q 300ALKATRESZ KOORDINÁTARENDszer ORIGÓJA
Q 403A SZERSZÁMGÉP ASZTALON
Q 546X:

A 510

R TRANS/

Q 646Y:

A 610

R ,

Q 746Z:

A 710

R ,

7Q 058PART

Q 200ALKATRESZ ANYAGANAK SZAMA

A 210

R PART/MATERL,

Q 400MEGMUNKALASI HELYEK

Q 4301. ELOMUNKALATLANOK

Q 5302. ELOMUNKALTAK

Q 6303 ELOONTOTTEK

Q1400A KERDESRE NEM KOTELEZO VALASZOLNI!

M 4 UNMACH,

M 5 SEMI,

M 6 CORED,

MOO

Q 800MUNKADARAB FELULETE

Q 8301. MEGMUNKALT

Q 9302. NYERS

M 8 SMOOTH,

M 9 ROUGH,

MOO

Q1000FORGACSOLASI PARAMETEREK MODOSITASA

Q11301. ELOTOLAS

A1110

R 8CORREC,

Q12302. FORDULATSZAM

Q1400

A1210

R ,

8Q 058SAFPOS

Q 300SZERSZAMCSERE HELYE

Q 341EGY PONTBAN

Q 441SIKBAN

J 3 *9

J 4 *10

9Q 058SAFPOS

Q 300SZERSZAMCSERE HELYE

Q 341EGY PONTBAN

<↑>

Q 441SIKBAN

< >

Q 600PONT KOORDINATAI:

Q 646X:

A 610

R SAFPOS/

Q 746Y:

A 710

R ,

Q 846Z:

A 810

R ,

G *11

10Q 058SAFPOS

Q 300SZERSZAMCSERE HELYE

Q 341EGY PONTBAN

< >

Q 441SIKBAN

< ↑ >

Q SIK Z KOORDINATAJA

A 610

R SAFPOS/

11Q 100MELYIK FEJEZETET KERED?

Q 4001. GEOMETRIAI MEGHATAROZASOK

Q 5002. TECHNOLOGIAI MEGHATAROZASOK

- Q 6003. LEHIVASOK VEGREHAJTASOK
Q 7004. EGYEB PROGRAMTECHNIKAI UTASITASOK
Q 8005. PROGRAMMANIPULACIOK
Q 9006. ALKATRESZPROGRAM VEGE
J 4 *12
J 5 *46
J 6 *68
J 7 *79
J 8 *89
J 9 *103
12Q 100MELYIK FEJEZETET KERED?
Q 300MEGHATAROZASOK
Q 4131. SIK
Q 5132. EGYEDI PONT
Q 6133. EGYENES
Q 7134. KOR
Q 8135. PONTMINTAZAT
Q 9136. KOORDINATARENDSZER TRANSFORMACIO
J 4 *13
J 5 *15
J 6 *20
J 7 *25
J 8 *29
J 9 *41
.
.
.
15Q 300A PONT ELHELYEZKEDIK:
Q 4201. SIKBAN
Q 5202. TERBEN
Q 6203. PONTMINTAZAT EGY ELEME
J 4 *16
J 5 *17
J 6 *18

.
.
.
17Q 058POINT

Q 300A PONT AZONOSITOJA:

A 3 6

R01 =POINT

Q 511KOORDINATAI:

Q 646X:

A 610

R /

Q 746Y:

A 710

R ,

Q 846Z:

A 810

R ,

G *19

.
.
.
19Q 200VALASZOLJ

Q 4021. UJABB PONTOT AKAROK MEGHATAROZNI

Q 6022. MAS GEOMETRIAI FEJEZETET KEREK

Q 8023. GEOMETRIAI RESZBOL KILEPEK

J 4 *15

J 6 *12

J 8 *11

.
.
.
.
.
F

KARTYAFILÉ FELHÍJTAS, A SZEGMENS SZÁMA: 0,
JAVÍTÁS: 8137

2 SSEG 0 0
3 DIAL 2
4 21HARD BE A KERT VALASZT
5 11HVISSZA < >
6 64H
7 1H/
8 1H
9 3H< >
10 1H
11 12H< >
12 23HVELYIK FEJEZETET KEREN
13 18H< >
14 18H< >
15 18H< >
16 18HVALASZOLJ
17 1H< >
18 42HA MEGADOTT AZONOSITO NEM KEZDŐDHEZ SZAMMAL
19 52HADJ U.I. AZONOSITOT MERT EZT MAR HASZNALTAD MAS HELYEN
20 49HA MEGADOTT AZONOSITOT NEM DEFINITLAD. VALASZOLJ
21 1H
22 11HRESZURAS< >
23 23HNYOMD MEG AZ 50H GUMBOT
24 11HJAVITAS < >
25 8H
26 18HIGEN < >
27 10HNEH < >
28 6HATANGI
29 4HINCR
30 2HAT
31 3HCLW
32 4HCLW
33 3HARC
34 4HTRH
35 6HINVERS
36 5HXYROT
37 1H1
38 1H2
39 3H0FF
40 2HON
41 0H
42 DIAL 3
1 43 0A
44
45 0F
46 DIAL 9
47 8LTH 4000.
48 SSEG 1 0
49 DIAL 5
50 6HARTNO
51 18HAZ ALKATRESZ NEVEO
52 1H<
53 1H>
54 6HACHIN
55 14HPOST-PROCESSOR
56 15HVN < >
57 15HNINCS < >
58 20HPOST-PROCESSOR NEVEO
59 6HNOPOST
60 6HCLPNT
61 23HCLTAPF-ROL LISTAT KEREN
62 9HREM KEREN
63 27HCLTAPF-ROL LISTAT NEM KERSZ
64 5HTRANS
65 36HALKATRESZ KOORDINATARENDSZER ORIGOJA
66 22HA SZERSZAMGEP ASZIALON
67 18HXO < >

68	18HY0	<	>
69	18H70	<	>
70	4HPART		
71	25HALKATRESZ ANYAGANAK SZAMA		
72	6HNMATERI		
73	19HMEGMUNKALASI HELYEK		
74	19H1. FLOMUNKALATLANOK		
75	15H2. FLOMUNKALTAK		
76	14H3. FLOONTOTTER		
77	25HA KERDESRE NEM KOTELEZO VALASZOLNI		
78	6HNMACH		
79	4HSEMT		
80	5HCORFD		
81	19HMUNKADARAB FEIULETE		
82	13H1. MEGMUNKALY		
83	8H2. NYERS		
84	6HSMOOTH		
85	5HROUGH		
86	15HFORHACSOLASI PARAMETEREK MODOSITASA		
87	11H1. FLOTOLAS		
88	15H2. FORDULATSZAM		
89	6HCORFEC		
90	17HFORHATSZAMMODOSITASHA VALASZI KEREK		
91	6HSAFPOS		
92	19HSZERZAMCSERE HELYE		
93	11HEGY PONTRAN		
94	6HSEKRA		
95	17HPONT KOORDINATAID		
96	19HSJK 2 KOORDINATAJAO		
97	23HMELYIK FEJEZETET KERED		
98	28H1. GEOMETRIAI MEGHATAROZASOK		
99	30H2. TECHNOLOGIAI MEGHATAROZASOK		
100	27H3. LEHIVASOK, VEGREHATJASOK		
101	36H4. FGYER PROGRAMTECHNIKAI UTASITASOK		
102	22H5. PROGRAMMANIPULACIOK		
103	24H6. ALKATRESZPROGRAM VEGF		
104	4HFINT		
105	0H		
106	DIAL 3		
1	107	12A	
	108	1	4 1 056 2 400 3 500 4 563 501 62
	109	3E	
	110	1T	
	111	0	
	112	3T	
	113	1	2 2
	114	3T	
	115	4	3 2
2	116	5A	
	117	3	1 -115000501
	118	1E	
	119	2T	
	120	1	1
3	121	7A	
	122	3	2 -21452 -315001500
	123	1E	
	124	2T	
	125	4	2
4	126	6A	
	127	4	3 1 -49999 1
	128	3E	
	129	3T	
	130	0	4 2
	131	3T	
	132	14	1 1
	133	3T	
	134	5	2 1
5	135	12A	

	136	2	4	5	058	6	100	7	349	8	4490362	1			
	137	3F													
	138	3I													
	139	3	6	2											
	140	3I													
	141	4	12	2											
	142	2I													
	143	14	2												
6	144	9A													
	145	3	3	-5	362	-21453	-31500	500							
	146	1F													
	147	2I													
	148	7	2												
7	149	5A													
	150	4	2	5	-4	1									
	151	1F													
	152	3I													
	153	1	8	1											
8	154	10A													
	155	1	3	9	500	3	600	4	663	601	62				
	156	3F													
	157	3I													
	158	14	5	1											
	159	3I													
	160	1	9	2											
	161	3I													
	162	4	10	2											
9	163	5A													
	164	3	1	-11500	601										
	165	1F													
	166	2I													
	167	8	1												
10	168	5A													
	169	3	1	-31500	1500										
	170	1F													
	171	2I													
	172	11	2												
11	173	4A													
	174	4	-19999	1											
	175	3F													
	176	3I													
	177	0	11	2											
	178	3I													
	179	14	5	1											
	180	3I													
	181	15	2	1											
12	182	9A													
	183	3	3	-5	462	-21453	-31500	1500							
	184	1F													
	185	2I													
	186	13	2												
13	187	4A													
	188	4	1	10	1										
	189	3F													
	190	3I													
	191	0	13	2											
	192	3I													
	193	14	5	1											
	194	3I													
	195	15	2	1											
14	196	5A													
	197	3	1	-11500	362										
	198	1F													
	199	2I													
	200	5	1												
15	201	14A													
	202	2	5	11	858	12	300	-6	361	13	414	-6	461	362	1
	203	3F													

204	31	3	16	2	
205	31	3	16	2	
206	31	4	18	2	
207	31	4	18	2	
208	31	4	18	2	
209	31	4	18	2	
210	31	4	18	2	
211	31	4	18	2	
212	31	4	18	2	
213	31	4	18	2	
214	31	4	18	2	
215	31	4	18	2	
216	31	4	18	2	
217	31	4	18	2	
218	31	4	18	2	
219	31	4	18	2	
220	31	4	18	2	
221	31	4	18	2	
222	31	4	18	2	
223	31	4	18	2	
224	31	4	18	2	
225	31	4	18	2	
226	31	4	18	2	
227	31	4	18	2	
228	31	4	18	2	
229	31	4	18	2	
230	31	4	18	2	
231	31	4	18	2	
232	31	4	18	2	
233	31	4	18	2	
234	31	4	18	2	
235	31	4	18	2	
236	31	4	18	2	
237	31	4	18	2	
238	31	4	18	2	
239	31	4	18	2	
240	31	4	18	2	
241	31	4	18	2	
242	31	4	18	2	
243	31	4	18	2	
244	31	4	18	2	
245	31	4	18	2	
246	31	4	18	2	
247	31	4	18	2	
248	31	4	18	2	
249	31	4	18	2	
250	31	4	18	2	
251	31	4	18	2	
252	31	4	18	2	
253	31	4	18	2	
254	31	4	18	2	
255	31	4	18	2	
256	31	4	18	2	
257	31	4	18	2	
258	31	4	18	2	
259	31	4	18	2	
260	31	4	18	2	
261	31	4	18	2	
262	31	4	18	2	
263	31	4	18	2	
264	31	4	18	2	
265	31	4	18	2	
266	31	4	18	2	
267	31	4	18	2	
268	31	4	18	2	
269	31	4	18	2	
270	31	4	18	2	
271	31	4	18	2	

	272	3I							
	273	4	27	2					
26	274	5A							
	275	3	1	-11500	653				
	276	1E							
	277	2I							
	278	25	1						
27	279	5A							
	280	3	1	-31500	753				
	281	1E							
	282	2I							
	283	28	2						
28	284	5A							
	285	4	2	-79999	1				
	286	1E							
	287	3I							
	288	1	29	1					
29	289	4A							
	290	1	0	753	10				
	291	3F							
	292	3I							
	293	14	21	1					
	294	3I							
	295	1	30	2					
	296	3I							
	297	4	31	2					
30	298	5A							
	299	3	1	-11500	753				
	300	1E							
	301	2I							
	302	29	1						
31	303	5A							
	304	3	1	-31500	1500				
	305	1E							
	306	2I							
	307	32	2						
32	308	5A							
	309	4	2	-79999	1				
	310	3F							
	311	3I							
	312	0	32	2					
	313	3I							
	314	14	21	1					
	315	3I							
	316	33	2	1					
33	317	10A							
	318	1	3	21	658	22	200	-8	252
	319	3F							
	320	1E							
	321	0							
	322	3I							
	323	1	34	2					
	324	1E							
	325	4	35	2					
34	326	5A							
	327	3	1	-11500	253				
	328	1E							
	329	2I							
	330	33	1						
35	331	7A							
	332	3	2	-21453	-31500	600			
	333	1E							
	334	2I							
	335	36	2						
36	336	8A							
	337	4	5	21	-4	23	-79999	1	
	338	1E							
	339	3I							

[illegible]

	408	2I							
	409	4B	2						
48	410	SA							
	411	4	2	-7	36	1			
	412	IF							
	413	3I							
	414	I	49	1					
49	415	14A							
	416	I	5	371000	381130	-81152	391230	-812521153	10
	417	3F							
	418	3I							
	419	14	33	1					
	420	3I							
	421	I	90	5					
	422	3I							
	423	4	50	5					
5	424	7A							
	425	4	4	-7	40	-79999	1		
	426	IF							
	427	3I							
	428	I	51	1					
5	429	4A							
	430	I		01283	10				
	431	3F							
	432	3I							
	433	15	33	1					
	434	3I							
	435	I	52	2					
	436	3I							
	437	4	53	2					
5	438	SA							
	439	3	1	4115001253					
	440	IF							
	441	2I							
	442	SI	1						
5	443	SA							
	444	3	1	-315001500					
	445	IF							
	446	2I							
	447	SA	2						
54	448	SA							
	449	4	2	-79999	1				
	450	3F							
	451	3I							
	452	A	54	2					
	453	3I							
	454	14	33	1					
	455	2I							
	456	SS	2	1					
55	457	16A							
	458	2	6	42 058	43 300	44 342	-6 361	45 442	-6 461 362 1
	459	3F							
	460	3I							
	461	3	56	2					
	462	3I							
	463	4	68	2					
	464	2I							
	465	74	2						
56	466	9A							
	467	3	3	-5 362	-21453	-31500	600		
	468	IF							
	469	2I							
	470	ST	2						
57	471	SA							
	472	4	2	42 -4	1				
	473	IF							
	474	3I							
	475	I	58	1					

58	476	124							
	477	1	4	46	600	18	646	19	746 29 846 653 10
	478	3F							
	479	3I							
	480	14	55	1					
	481	3I							
	482	1	59	2					
	483	3I							
	484	4	60	2					
59	485	5A							
	486	3	1	-11500	653				
	487	1F							
	488	2I							
	489	58	1						
60	490	5A							
	491	3	1	-31500	753				
	492	1F							
	493	2I							
	494	61	2						
61	495	4A							
	496	4	10059	1					
	497	1E							
	498	3I							
	499	1	62	1					
62	500	4A							
	501	1	0	753	10				
	502	3F							
	503	3I							
	504	14	55	1					
	505	3I							
	506	1	63	2					
	507	3I							
	508	4	64	5					
63	509	5A							
	510	3	1	-11500	753				
	511	1F							
	512	2I							
	513	62	1						
64	514	5A							
	515	4	2	-79999	1				
	516	1E							
	517	3I							
	518	1	65	1					
65	519	4A							
	520	1	0	853	10				
	521	3F							
	522	3I							
	523	14	55	1					
	524	3I							
	525	1	66	2					
	526	3I							
	527	4	75	2					
66	528	5A							
	529	3	1	-11500	853				
	530	1E							
	531	2I							
	532	65	1						
67	533	5A							
	534	4	2	-79999	1				
	535	3F							
	536	3I							
	537	0	67	2					
	538	3I							
	539	14	55	1					
	540	3I							
	541	76	2	1					
68	542	5A							
	543	3	3	-5.462	-21453	-31500	600		

[illegible]

	612	76	1							
7A	613	12A								
	614	2	4	-5	462	-21453	-31500	4915001500	1	
	615	2F								
	616	3I								
	617	14	76	1						
	618	2I								
	619	84	1							
7B	620	12A								
	621	2	4	-5	562	-21453	-31500	5015001500	1	
	622	2F								
	623	3I								
	624	14	76	1						
	625	2I								
	626	85	1							
8	627	12A								
	628	2	4	-5	662	-21453	-31500	5115001500	1	
	629	2F								
	630	3I								
	631	14	76	1						
	632	2I								
	633	86	1							
81	634	12A								
	635	2	4	-5	762	-21453	-31500	5215001500	1	
	636	2F								
	637	3I								
	638	14	76	1						
	639	2I								
	640	87	1							
82	641	12A								
	642	2	4	-5	862	-21453	-31500	5315001500	1	
	643	2F								
	644	3I								
	645	14	76	1						
	646	2I								
	647	88	1							
83	648	9A								
	649	3	3	-5	962	-21453	-315001500			
	650	1F								
	651	2I								
	652	89	2							
84	653	2A								
	654	-2	1							
	655	0F								
85	656	2A								
	657	-6	4							
	658	0F								
86	659	2A								
	660	-8	7							
	661	0F								
87	662	2A								
	663	-5	56							
	664	0F								
88	665	2A								
	666	-9	42							
	667	0F								
89	668	4A								
	669	4	1	65	1					
	670	3F								
	671	3I								
	672	0	89	2						
	673	3I								
	674	14	76	1						
	675	3I								
	676	89	2	2						
90	677	4A								
	678	4	1	0	1					
	679	3F								

[illegible]

	748	6	5	3					
	749	3I							
	750	7	6	3					
	751	3I							
	752	8	7	3					
	753	3I							
	754	9	8	3					
	755	3I							
	756	14	1	1					
	757	2I							
	758	2	2						
2	759	5A							
	760	3	1	-11500	462				
	761	1F							
	762	2I							
	763	1	1						
3	764	10A							
	765	2	3	-5	462	-31500	21500	1500	1
	766	2F							
	767	3I							
	768	14	1	1					
	769	2I							
	770	9	1						
4	771	10A							
	772	2	3	-5	562	-31500	31500	1500	1
	773	2F							
	774	3I							
	775	14	1	1					
	776	2I							
	777	18	1						
5	778	10A							
	779	2	3	-5	662	-31500	41500	1500	1
	780	2F							
	781	3I							
	782	14	1	1					
	783	2I							
	784	29	1						
6	785	10A							
	786	2	3	-5	762	-31500	51500	1500	1
	787	2F							
	788	3I							
	789	14	1	1					
	790	2I							
	791	14A	1						
7	792	10A							
	793	2	3	-5	862	-31500	61500	1500	1
	794	2F							
	795	2I							
	796	14	1	1					
	797	2I							
	798	193	1						
8	799	10A							
	800	2	3	-5	962	-31500	71500	1500	1
	801	2F							
	802	3I							
	803	14	1	1					
	804	2I							
	805	194	1						
9	806	12A							
	807	1	4	8	058	9	400	-12	546
	808	3F							
	809	3I							
	810	14	1	1					
	811	3I							
	812	1	10	2					
	813	3I							
	814	4	11	2					
10	815	5A							

[illegible]

[illegible]

30	952	5A							
	953	3	1	-11500	653				
	954	1F							
	955	2I							
	956	29	1						
31	957	5A							
	958	3	1	-31500	753				
	959	1F							
	960	2I							
	961	32	2						
32	962	4A							
	963	4	10000	1					
	964	1F							
	965	3I							
	966	1	33	1					
33	967	4A							
	968	1	0 753	10					
	969	2F							
	970	3I							
	971	14	23	1					
	972	3I							
	973	1	34	2					
	974	2I							
	975	4	35	2					
34	976	5A							
	977	3	1	-11500	753				
	978	1F							
	979	2I							
	980	33	1						
35	981	5A							
	982	3	1	-31500	1500				
	983	1F							
	984	2I							
	985	36	2						
36	986	5A							
	987	4	2	-79094	1				
	988	2F							
	989	2I							
	990	0	36	2					
	991	3I							
	992	14	23	1					
	993	3I							
	994	73	2	1					
37	995	14A							
	996	1	5	16 058	17 300	18 311	-14 356	-21453	357 0
	997	5F							
	998	3I							
	999	14	18	1					
	1000	3I							
	1001	1	38	2					
	1002	3I							
	1003	2	39	2					
	1004	3I							
	1005	3	40	2					
	1006	3I							
	1007	4	41	2					
38	1008	5A							
	1009	3	1	-11500	357				
	1010	1F							
	1011	2I							
	1012	37	1						
39	1013	7A							
	1014	3	2	-31500	-151500	357			
	1015	1F							
	1016	2I							
	1017	37	1						
40	1018	7A							
	1019	3	2	-31500	-151500	357			

	1020	1F							
	1021	2I							
	1022	37	1						
41	1023	5A							
	1024	3	1	-31500	811				
	1025	1F							
	1026	2I							
	1027	42	2						
42	1028	7A							
	1029	4	49999	-18	16	-4	1		
	1030	1F							
	1031	3I							
	1032	1	43	1					
43	1033	12A							
	1034	1	4	19	511	-10	646	-11	746
	1035	3F							
	1036	3I							
	1037	14	37	1					
	1038	3I							
	1039	1	44	2					
	1040	3I							
	1041	4	45	2					
44	1042	5A							
	1043	3	1	-11500	653				
	1044	1F							
	1045	2I							
	1046	43	1						
45	1047	5A							
	1048	3	1	-31500	753				
	1049	1F							
	1050	2I							
	1051	46	2						
46	1052	4A							
	1053	4	19999	1					
	1054	1F							
	1055	3I							
	1056	1	47	1					
47	1057	4A							
	1058	1	0	753	10				
	1059	3F							
	1060	3I							
	1061	14	37	1					
	1062	3I							
	1063	1	48	2					
	1064	3I							
	1065	4	49	2					
48	1066	5A							
	1067	3	1	-11500	753				
	1068	1F							
	1069	2I							
	1070	47	1						
49	1071	5A							
	1072	3	1	-31500	853				
	1073	1F							
	1074	2I							
	1075	50	2						
50	1076	5A							
	1077	4	2	-79999	1				
	1078	1F							
	1079	3I							
	1080	1	51	1					
51	1081	4A							
	1082	1	0	853	10				
	1083	3F							
	1084	3I							
	1085	14	37	1					
	1086	3I							
	1087	1	52	2					

[illegible]

	1156	31			
	1157	3	64	2	
	1158	31			
	1159	4	67	2	
62	1160	5A			
	1161	3	1	-11500	557
	1162	1F			
	1163	21			
	1164	61	1		
63	1165	7A			
	1166	3	2	-31500 -151500	557
	1167	1F			
	1168	21			
	1169	61	1		
64	1170	9A			
	1171	3	2	-31500 -171500 -191553	557
	1172	1F			
	1173	21			
	1174	65	1		
65	1175	4A			
	1176	2	0	557	1
	1177	3F			
	1178	31			
	1179	14	55	1	
	1180	31			
	1181	15	105	1	
	1182	21			
	1183	66	2		
66	1184	4A			
	1185	3	1	-20	557
	1186	1F			
	1187	21			
	1188	61	1		
67	1189	5A			
	1190	4	1	-31500	700
	1191	1F			
	1192	21			
	1193	68	2		
68	1194	4A			
	1195	4	10000	1	
	1196	1F			
	1197	31			
	1198	1	60	1	
69	1199	8A			
	1200	1	2	21 700	8 752 753 10
	1201	3F			
	1202	31			
	1203	14	55	1	
	1204	31			
	1205	1	70	2	
	1206	31			
	1207	4	71	2	
70	1208	5A			
	1209	3	1	-11500	753
	1210	1F			
	1211	21			
	1212	69	1		
71	1213	5A			
	1214	3	1	-315001500	
	1215	1F			
	1216	21			
	1217	72	2		
72	1218	5A			
	1219	4	2	-79999	1
	1220	3F			
	1221	31			
	1222	0	72	2	
	1223	31			

[illegible]

4. Függelék

AZ AIR PROGRAM FORTRAN LISTÁJA


```
LN 0002      OVERLAY(2)
LN 0003      COMMON/DIALCO/ YS,IA,ISV,IAY,IST,IPL(10),J,IETM
LN 0004      COMMON/DIALTAB/ ITAB(5000)
LN 0005      COMMON/BLOK1/ IR,IXY,IX,IY,ICXY,IAN,NIP,INP,IYOV,IP
LN 0006      COMMON/BLOK2/ IDSZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),IUPUF(9)
LN 0007      F,IIPR(1000),ISNPP(50)
LN 0008      COMMON/BLOK3/ IFLAG1,IFLAG2,IFLAG3,IFLAG4,IFLAG5,IFLAG6,IFLAG7
LN 0009      COMMON/BLOK4/ ISPACF,ISP,ISZAM(10),IFORM(3)
LN 0010      COMMON/BLOK5/ IXCY,IXC7,IYCY,IYC7
LN 0011      COMMON/BLOK6/ KUT,IAZT,IUN,KSSN(2),IPUF(9)
LN 0012      COMMON/BLOK7/ IKAR(R)
LN 0013      COMMON/BLOK8/ IC
LN 0014      COMMON/BLOK9/ IASSZ,IUSSZ,IAZON,INAM(2)
LN 0015      COMMON/BLOK 10/ MKT
LN 0016      J=INPAG(1)
LN 0017      CALL FOPR
LN 0018      STOP
LN 0019      END
```

USASI FORTRAN DIAGNOSTIC RESULTS FOR AIR

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

DIALCO	DIALTAB	BLOK1	BLOK2
BLOK3	BLOK4	BLOK5	BLOK6
BLOK7	BLOK8	BLOK9	BLOK10

HSAST-FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001      SUBROUTINE FOUR
LN 0002      COMMON/BLOK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP,/BLOK2/
LN 0003      IINDZIM(15,9),IPUF(R),IAZPUF(2,50),IUTOMB(30),IUPUF(9),IPPR(1000),
LN 0004      2ISNPP(50)
LN 0005      COMMON/BLOK3/IFLAG1,IFLAG2,IFLAG3,IFLAG4,IFLAG5,IFLAG6,IFLAG7,
LN 0006      3/BLOK4/ISPACE,ISP,ISZAM(10),IFORM(3),/BLOK5/IXCY,IXC7,IYCY,IYC7,
LN 0007      4/BLOK6/KUT,IAZT,IUH,KSSN(2),IFPUF(9)
LN 0008      COMMON/BLOK7/IKAR(R)
LN 0009      COMMON/BLOK8/IC
LN 0010      COMMON/BLOK9/ IASSZ,IUSSZ,IAZON,INAM(2)
LN 0011      COMMON/BLOK 10/ MKI
LN 0012      COMMON/BLOK11/I7SURF
LN 0013      DATA ISPACE/RM /,ISP/1H /,ISZAM/1,2,3,4,5,6,7,8,9,0/,
LN 0014      FIFORM/4H 0 04H 0 04H 0 /
LN 0015      DATA IKAR(1)/RM1234567R/,IKAR(2)/RM910 /
LN 0016      DATA I7SURF /RM7SURF /
LN 0017      IOP=1
LN 0018      INP=1
LN 0019      IOP=1
LN 0020      IAN=1
LN 0021      KSSN(1)=0
LN 0022      KSSN(2)=0
LN 0023      1 CALL NUL
LN 0024      2 CALL OUT 1
LN 0025      CALL KIIRAS
LN 0026      3 CALL IN
LN 0027      CALL KIIRAS
LN 0028      IR=IANAL(IC)
LN 0029      GOTO(1,7,2,5,4,9,10,11,12,13,14,15),IC
LN 0030      7 CALL OUT 1
LN 0031      CALL KIIRAS
LN 0032      IR=IANAL(IC)
LN 0033      GOTO(1,4),IC
LN 0034      4 CALL ISTAT
LN 0035      IR=IANAL(IC)
LN 0036      GOTO(2,5),IC
LN 0037      5 CALL KONVS
LN 0038      CALL OUT 2
LN 0039      CALL KIIRAS
LN 0040      CALL IN
LN 0041      CALL KIIRAS
LN 0042      IR=IANAL(IC)
LN 0043      GOTO(1,6,16),IC
LN 0044      6 CALL IALKP
LN 0045      GOTO(1,8),ITOV
LN 0046      8 CALL PRT
LN 0047      GOTO 17
LN 0048      9 CALL TUTLK
LN 0049      GOTO 2
LN 0050      10 CALL IALK
LN 0051      GOTO 2
LN 0052      11 CALL IPUK
LN 0053      GOTO 2
LN 0054      12 CALL TUTORL

```


USASI REPORTING AIDS

USASI FORTRAN(2.0)/MASTER

INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF

09/04/72

PAGE 002

```
LN 0055      GOTO 2
LN 0056      13 CALL THREW
LN 0057      GOTO 2
LN 0058      14 CALL KOSTKI
LN 0059      GOTO 2
LN 0060      15 CALL INTKI
LN 0061      GOTO 2
LN 0062      16 CALL KSTAD
LN 0063      GOTO 4
LN 0064      17 CONTINUE
LN 0065      RETURN
LN 0066      END
```

USASI FORTRAN DIAGNOSTIC RESULTS FOR FORR

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1	BLOCK2	BLOCK3	BLOCK4
BLOCK5	BLOCK6	BLOCK7	BLOCK8
BLOCK9	BLOCK10	BLOCK11	


```

USASI FORTRAN(2,4)/HASTED  INTEGER WORD SIZE = 2 * * OPTION 15 OFF * 0 OPTION 15 OFF 09/04/72 PAGE 001

LN 0001      SUBROUTINE TALKP
LN 0002      COMMON/BLOCK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003      COMMON/BLOCK2/IDSZIN(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(50),
LN 0004      IUPUF(9),IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLOCK4/ISPACE,ISP,ISZAN(10),IFORM(3)
LN 0006      COMMON/BLOCK6/KUT,IAZI,IUH,KSSN(2),IFPUF(9)
LN 0007      COMMON/BLOCK8/IC
LN 0008      ISNPP(INP) = IP
LN 0009      IPPR(ITP) = IUH
LN 0010      DO1 I=1,IUH
LN 0011      K=IP+1
LN 0012      1 IPPR(K)=IUPUF(I)
LN 0013      IP = IP + IUH + 1
LN 0014      IF (IAZI.EQ.ISPACE) GOTO 2
LN 0015      IAZPUF(1,IAN)=INP
LN 0016      IAZPUF(2,IAN)=IAZI
LN 0017      IAN=IAN+1
LN 0018      2 INP=INP+1
LN 0019      RETURN
LN 0020      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR TALKP

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1      BLOCK2      BLOCK4      BLOCK6
BLOCK8

```

USASI FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , * OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001      SUBROUTINE NULL
LN 0002      COMMON/BLOCK2/IDSZIM(16,0),IPUF(R),IAZPUF(2,50),IUTOMB(30),
LN 0003      FIIPUF(0),IPPR(1000),ISNPP(50)
LN 0004      COMMON/BLOCK4/ISPACE,ISP,ISZAM(10),IFORM(3)
LN 0005      COMMON/BLOCK9/ IASSZ,IUSSZ,IAZON,INAM(2)
LN 0006      DO1 T=1,16
LN 0007      DO1 K=1, 0
LN 0008      I IDSZIM(T,K)=ISPACE
LN 0009      RETURN
LN 0010      END
```

USASI FORTRAN DIAGNOSTIC RESULTS FOR NULL

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK2	BLOCK4	BLOCK9
--------	--------	--------

USAST FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001      SUBROUTINE KSTAR
LN 0002      CHARACTER ISIK(8),KONVP(8)
LN 0003      COMMON/BLOK1/IR,IXY,IX,IV,ICXY,IAN,NIP,INP,ITOV,IP
LN 0004      COMMON/BLOK2/IDSZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),IUPUF(9)
LN 0005      F,IPRR(1000),ISNRR(50)
LN 0006      COMMON/BLOK6/KUT,IAZI,IUM,KSSN(2),IFPUF(9)
LN 0007      COMMON/BLOK11/I2SURF
LN 0008      EQUIVALENCE (ISIK,I2SURF)
LN 0009      EQUIVALENCE (IUPUF,KONVP)
LN 0010      DO1 I=1,5
LN 0011      IF(KONVP(I).NE.ISIK(I))GOTO 2
LN 0012      1 CONTINUE
LN 0013      KSSN(2)=INP-1
LN 0014      RETURN
LN 0015      2 KSSN(1)=INP-1
LN 0016      RETURN
LN 0017      END

```

USAST FORTRAN DIAGNOSTIC RESULTS FOR KSTAR

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOK1	BLOK2	BLOK6	BLOK11
-------	-------	-------	--------

022 09/04/72

USAST-FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF * 0 OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001      SUBROUTINE INTKI
LN 0002      COMMON/BLOCK1/IR,IXY,IX,IY,ICXY,IAM,NIP,INP,ITOV,IP
LN 0003      COMMON/BLOCK2/IOSZTM(16,9),TPUF(8),IAZPUF(2,59),IUTOWB(30),IUPUF(9)
LN 0004      F=TPPR(1000),ISNPP(50)
LN 0005      CALL NULL
LN 0006      ISNPP(INP)=TP
LN 0007      TPRP(TP)=R
LN 0008      DO1 I=1, R
LN 0009      K=TP+1
LN 0010      1 TPRP(K)=IUPF(I)
LN 0011      TP=TP+9
LN 0012      INP=INP+1
LN 0013      RETURN
LN 0014      END
```

USAST-FORTRAN DIAGNOSTIC RESULTS FOR INTKI

NO ERRORS

THE FOLLOWING ARE COMMON-BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1 BLOCK2

```

USAST FORTRAN2,41/MASTER    INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF    09/04/72    PAGE 001

LN 0001      SUBROUTINE PRT
LN 0002      COMMON/BLOCK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003      COMMON/BLOCK2/IDSZTW(16,9),TUPF(8),IAZPUE(2,50),IUTOMB(30),TUPUF(9)
LN 0004      F,TPPR(1000),ISNPP(50)
LN 0005      1 FORMAT(1H) //16HAL KATRESZPROGRAM)
LN 0006      WRITE(2,1)
LN 0007      2 FORMAT(1HH,13,5X,888)
LN 0008      IF IN=INP-1
LN 0009      DO3 I=1, IFIN
LN 0010      IUH=ISNPP(I)
LN 0011      L=IUH+1
LN 0012      IUHO=TPPR(IUH)
LN 0013      IVEG=IUH+IUHO
LN 0014      WRITE(2,2) I,(TPPR(IL),IL=L,IVEG)
LN 0015      3 CONTINUE
LN 0016      RETURN
LN 0017      END

USAST FORTRAN DIAGNOSTIC RESULTS FOR PRT

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1    BLOCK2

```



```
74 09/04/72
HSA5T FORTRAN(2.0)/MASTER  INTEGR WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

LN 0001      SHRODITNF NUL
LN 0002      COMMON/BLOCK2/IDSZIN(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),
LN 0003      F(IUPUF(9),IPPR(100),ISNPP(50)
LN 0004      COMMON/BLOCK3/IFLAG1,IFLAG2,IFLAG3,IFLAG4,IFLAG5,IFLAG6,IFLAG7
LN 0005      COMMON/BLOCK4/ISPACE,ISP,TSZAM(10),IFORM(3)
LN 0006      COMMON/BLOCK6/KUT,IAZT,IUH,KSSH(2),IFPUF(9)
LN 0007      COMMON/BLOCK9/ TASSZ,IUSSZ,IAZON,TNAM(2)
LN 0008      CALL NULL
LN 0009      DO I=1,0
LN 0010      1 IUPUF(I)=ISPACE
LN 0011      DOZ L=1,30
LN 0012      2 IUTOMB(L)= ISPACE
LN 0013      IFLAG1=0
LN 0014      IFLAG2=0
LN 0015      IFLAG3=0
LN 0016      IFLAG4=0
LN 0017      KUT=1
LN 0018      IAZT=ISPACE
LN 0019      TASSZ=0
LN 0020      IUSSZ=0
LN 0021      RETURN
LN 0022      END

HSA5T FORTRAN DIAGNOSTIC RESULTS FOR NUL

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
BLOCK2      BLOCK3      BLOCK4      BLOCK6
BLOCK9
```



```
USASI FORTRAN(2.4)/MASTER  INTEGER WORD SIZE = 2 * * OPTION IS OFF * 0 OPTION IS OFF 09/04/72 PAGE 001

LN 0001      SUBROUTINE      ICURS
LN 0002      COMMON/BLOCK1/IY,IX,IX,IX,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003      IYA=IXY/100
LN 0004      IYA=IXY-100*IXA
LN 0005      IX=IXA*1
LN 0006      IY=IYA*1
LN 0007      RETURN
LN 0008      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR ICURS

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1
```

USAST-FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001        SUBROUTINE    MISTAK (IST, IHSZ)
LN 0002        1 FORMAT(26MHPPROGRAMHIRA    MIRA SZAMA:5X,13)
LN 0003        2 FORMAT(42MHPPROGRAMHIRA    A PROGRAM NEM FUTHAT TOVARA/
LN 0004        F12HHHIRA SZAMA:5X,13)
LN 0005        IF (IST.F0.1) GOTO 8
LN 0006        WRITE(2,1) IHSZ
LN 0007        RETURN
LN 0008        8 WRITE(2,2) IHSZ
LN 0009        STOP
LN 0010        END
```

USAST FORTRAN DIAGNOSTIC RESULTS FOR MISTAK

NO ERRORS

USASI FORTRAN2.0/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001      SUBROUTINE      OUT2
LN 0002      COMMON/BLOCK1/TR,IXY,IX,IY,ICXY,IAN,NIR,INP,ITOV,IP
LN 0003      COMMON/BLOCK2/ISZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),
LN 0004      FI(IPUF(9),IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLOCK6/ISPACE,ISP,ISZAM(10),IFORM(3)
LN 0006      COMMON/BLOCK6/KUT,IAZT,IUH,KSSN(2),IFPUF(9)
LN 0007      COMMON/BLOCK8/IC
LN 0008      ISZIM(1,1) = INP
LN 0009      DO3 I=1,9
LN 0010      3 ISZIM(16,I)=ISPACE
LN 0011      IF(IUH,AT,8) GOTO 15
LN 0012      M = IUH
LN 0013      IX = 16
LN 0014      4 DO1 I=1,M
LN 0015      1 ISZIM(IX,I)= IPUF(I)
LN 0016      1 CONTINUE
LN 0017      IF(IUH,AT,8) GOTO 16
LN 0018      RETURN
LN 0019      16 N=IUH-M
LN 0020      IX = 16
LN 0021      DO2 I=1,N
LN 0022      K=M+I
LN 0023      2 ISZIM(IX,I)=IUBUF(K)
LN 0024      RETURN
LN 0025      15 M = 6
LN 0026      IX = 15
LN 0027      GOTO 4
LN 0028      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR OUT2

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1	BLOCK2	BLOCK4	BLOCK6
BLOCK8			

USAST FORTRAN(2.0)/MASTER INTEGR WORD SIZE = 2 * OPTION IS OFF * OPTION IS OFF 09/06/72 PAGE 001

```

LN 0001      SUBROUTINE IN
LN 0002      COMMON/BLOK1/IR,IX,IXY,IXY,TCXY,IAN,NIP,INP,ITOV,IP
LN 0003      COMMON/BLOK2/IOSZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),
LN 0004      ITHPUF(4),IPPR(1A00),ISNPP(50)
LN 0005      COMMON/BLOK4/ISPACE,ISP,ISZAM(10),IFORM(3)
LN 0006      COMMON/BLOK8/IC
LN 0007      DO1 I=1,9
LN 0008      1 ITHUF(I)= ISPACE
LN 0009      ITP = JAKP(IR,NIP,L)
LN 0010      IF(ITYP,EQ,3) GOTO4
LN 0011      NTP = L
LN 0012      NTP = JAKP(IR,NIP,L)
LN 0013      IF ( NTP,EQ,1) GOTO2
LN 0014      IXY = JAKP(IR,NTP=1,L)
LN 0015      IF ( NTP,EQ,6) GOTO5
LN 0016      IF ( NTP,EQ,10) GOTO6
LN 0017      IF (NIP,LE,62) GOTO 5
LN 0018      CALL MISTAK(1,4)
LN 0019      RETURN
LN 0020      4 CALL MISTAK(1,6)
LN 0021      RETURN
LN 0022      2 READ(11,1001) TCXY
LN 0023      1001 FORMAT (I4)
LN 0024      IXY=TCXY
LN 0025      CALL ICHRS
LN 0026      ITH=IOSZT(IXY)
LN 0027      IFO=IFORM(1)
LN 0028      GO TO 10
LN 0029      5 N=1
LN 0030      3 IFO=IFORM(N)
LN 0031      J=IOSZT(NIP)
LN 0032      READ(11,1002) TCXY, (IPUF(I),I=1,J)
LN 0033      1002 FORMAT (I4,A8)
LN 0034      IF (TCXY,EQ,IXY) GOTO 8
LN 0035      CALL MISTAK(10,5)
LN 0036      GOTO 8
LN 0037      6 N=2
LN 0038      GOTO 3
LN 0039      8 IXY=TCXY
LN 0040      CALL ICHRS
LN 0041      ITH=IOSZT(IXY)
LN 0042      DO 9 I=1,J
LN 0043      K=ITH+I
LN 0044      9 IOSZIM(IX,K)=IPUF(I)
LN 0045      10 IOSZIM(IX,ITH)=IFO
LN 0046      RETURN
LN 0047      END

```

USAST FORTRAN DIAGNOSTIC RESULTS FOR IN

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOK1	BLOK2	BLOK4	BLOK8
-------	-------	-------	-------

USASY FORTRAN12.01/MASTER INTEGER WORD SIZE = 2 ; * OPTION IS OFF ; 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001 SUBROUTINE KONVS
LN 0002 COMMON/BLK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003 COMMON/BLK2/IDSZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMR(30),
LN 0004 IIPUF(6),IPPR(1000),ISNPP(50)
LN 0005 COMMON/BLK4/ISPACE,ISP,ISZAM(10),IFORM(13)
LN 0006 COMMON/BLK6/KUT,IAZT,IUH,KSSN(2),IFPUF(9)
LN 0007 CHARACTER KONV(240),KONVP(72),KSP,KIAZ(8),KIAZT(8)
LN 0008 EQUIVALENCE (IUTOMR,KONV),(IUPUF,KONVP),(IAZ,KIAZ),(IAZT,KIAZT)
LN 0009 KSP=ISPACE
LN 0010 IREF=1
LN 0011 IERY=-10
LN 0012 J=MOVFS(IFPUF,IREF,IERV)
LN 0013 IF(IUTOMR(2).NE.IFPUF(1)) GOTO 8
LN 0014 L=0
LN 0015 IAZ=IUTOMR(1)
LN 0016 DOA I=1,8
LN 0017 IF(KIAZ(I).EQ.KSP) GOTO 4
LN 0018 L=L+1
LN 0019 KIAZT(I)=KIAZ(I)
LN 0020 4 CONTINUE
LN 0021 8 K=0
LN 0022 IF(INIP.EQ.62) GOTO 2
LN 0023 IVEG=(KUT-1)*8
LN 0024 DO I=1, IVEG
LN 0025 IF(KONV(I).EQ.KSP) GOTO 1
LN 0026 K=K+1
LN 0027 KONVP(K)=KONV(I)
LN 0028 1 CONTINUE
LN 0029 IUH=IOSZT(K)
LN 0030 RETURN
LN 0031 2 DO 3 I=1,16
LN 0032 IF(KONV(I).EQ.KSP) GOTO 3
LN 0033 K=K+1
LN 0034 KONVP(K)=KONV(I)
LN 0035 3 CONTINUE
LN 0036 IUH=0
LN 0037 RETURN
LN 0038 END

```

USASY FORTRAN DIAGNOSTIC RESULTS FOR KONVS

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLK1 BLK2 BLK4 BLK6

USASI FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 * OPTION IS OFF * OPTION IS OFF 09/04/72 PAGE 001

LN 0001 INTEGER FUNCTION IOSZT(NOP)
LN 0002 IOSZT=(NOP+7)/8
LN 0003 RETURN
LN 0004 END

USASI FORTRAN DIAGNOSTIC RESULTS FOR IOSZT

NO ERRORS


```

USAST FORTRAN 2.0/MASTED      INTEGER WORD SIZE = 2 * * OPTION IS OFF , 0 OPTION IS OFF      09/04/72      PAGE 001

LN 0001      SUBROUTINE KIIRAS
LN 0002      COMMON/BLOCK 1/IR,IXY,IX,IY,ICXY,IAN,INP,INP,ITOV,IP
LN 0003      COMMON /BLOCK 2/ IDSZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),
LN 0004      IIPUF(9),IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLOCK 10/ MKI
LN 0006      1 FORMAT(1H) ////)
LN 0007      2 FORMAT(1HH,////// 25X,74H*****
LN 0008      ***** +/16(25X,1H*,9AB,1H*/),25X,74H
LN 0009      *****
LN 0010      N*****
LN 0011      IF(MKI.F0.1) GOTO 4
LN 0012      WRITE(2,1)
LN 0013      MKI=1
LN 0014      3 WRITE(2,2)((IDSZIM(I,K),K=1,9),I=1,16)
LN 0015      RETURN
LN 0016      4 MKI=0
LN 0017      GOTO 3
LN 0018      END

USAST FORTRAN DIAGNOSTIC RESULTS FOR KIIRAS

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
BLOCK1      BLOCK2      BLOCK10

```

12. 09/04/72

USAST-FORTRAN(2.0)/MASTER

INTEGER WORD SIZE = 2 * * OPTION IS OFF * 0 OPTION IS OFF

09/04/72

PAGE 001

```

LN 0001      INTEGER FUNCTION TANAL (IC)
LN 0002      CHARACTER KSYMR(8),KAZON(8),INUM(16),KSP
LN 0003      COMMON/RIOK1/IR,IXY,IX,TY,TCXY,TAN,NTP,INP,ITOV,IP,/BLOK2/
LN 0004      FIOS7IM(16,0),IPIUF(8),IAZPUF(2,50),IUTOMB(30),IUPUF(9),IPPR(1000),
LN 0005      2TENDP(50)
LN 0006      COMMON/RIOK3/IFLAG1,IFLAG2,IFLAG3,IFLAG4,IFLAG5,IFLAG6,IFLAG7
LN 0007      COMMON/RIOK4/ISPACE,ISP,ISZAM(10),IFORM(3)
LN 0008      COMMON/RIOK7/IKAR(8)
LN 0009      COMMON/RIOK9/ TASS7,IUSSZ,IAZON,INAM(2)
LN 0010      EQUIVALENCE (IKAR,INUM)
LN 0011      EQUIVALENCE (TSYMR,KSYMR),(IAZON,KAZON)
LN 0012      ITYP=JAKP(IR,1,1)
LN 0013      IF (ITYP.EQ.3) GOTO 1
LN 0014      IF (ITYP.EQ.2) GOTO 2
LN 0015      IF (ITYP.EQ.1) GOTO 4
LN 0016      ICY=JFLP(IR,1,1,L)
LN 0017      IF (ICY.EQ.1) GOTO 6
LN 0018      GOTO 2R
LN 0019      1 INT=1
LN 0020      4 IANAL=JFLP(IR,INT,1,L)
LN 0021      IC=JFLP(IR,INT,2,L)
LN 0022      IANAL=ICSAT(IANAL)
LN 0023      IANAL=ISURP(IANAL)
LN 0024      2R RETURN
LN 0025      2 ICI=NFP(IR)
LN 0026      JF=ICI-1
LN 0027      DO3 I=1,JF
LN 0028      K=I
LN 0029      JFL=JFLP(IR,K,1,L)
LN 0030      JFL=JFL+1
LN 0031      IF (IX.EQ. JFL) GOTO 5
LN 0032      3 CONTINUE
LN 0033      INT=ICL
LN 0034      GOTO 4
LN 0035      27 IFLAG2=1
LN 0036      GOTO 6
LN 0037      2R IF (IFLAG2.EQ.0) GOTO 27
LN 0038      IF (IX.EQ.15) GOTO 7
LN 0039      INT=3
LN 0040      ITOV=JFLP(IR,INT,3,L)
LN 0041      GOTO 4
LN 0042      7 INT=2
LN 0043      8 IANAL=JFLP(IR,INT,2,L)
LN 0044      IC=JFLP(IR,INT,3,L)
LN 0045      IANAL=ICSAT(IANAL)
LN 0046      IANAL=ISURP(IANAL)
LN 0047      GOTO 2R
LN 0048      5 INT=1
LN 0049      GOTO 8
LN 0050      6 INT=1
LN 0051      GOTO 8
LN 0052      9 ICY=JFLP(IR,1,1,L)
LN 0053      ICY=ICY+1
LN 0054      IF (ICY.EQ.15) GOTO 10

```



```

USASI-FORTRAN(2.0)/MASTER  INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF  09/04/72  PAGE 002

LN 0055      11 DO 12 I=1,R
LN 0056      IF(IPUF(I).NE.ISPACE)GOTO 13
LN 0057      12 CONTINUE
LN 0058      J=1
LN 0059      23 IFL=NPR(ITR)
LN 0060      DO 24 I=2,IFL
LN 0061      KY = I
LN 0062      JFL=JFLP(IP,KY+1,L)
LN 0063      IF(JFL.FQ.)GOTO 5
LN 0064      24 CONTINUE
LN 0065      CALL MISTAK(1,3)
LN 0066      10 IF(IX.NE.ICY)GOTO 11
LN 0067      GOTO 6
LN 0068      13 IF(NIP.NE.6)GOTO 21
LN 0069      IAZON=ISPACE
LN 0070      KSP=ISPACE
LN 0071      L=0
LN 0072      KSYM=IPUF(1)
LN 0073      DO 14 I=1,R
LN 0074      IF(KSYM(I).EQ.KSPI)GOTO 14
LN 0075      L=L+1
LN 0076      KAZON(L)=KSYM(I)
LN 0077      14 CONTINUE
LN 0078      DO 15 I=1,TAN
LN 0079      II=I
LN 0080      IF(IAZON.EQ.IAZPUF(2,II))GOTO 19
LN 0081      15 CONTINUE
LN 0082      IF(IFLAG1.FQ.0) GOTO 16
LN 0083      20 J=3
LN 0084      GOTO 23
LN 0085      19 IF(IFLAG1.FQ.0) GOTO 20
LN 0086      21 J=4
LN 0087      IFLAG1=1
LN 0088      GOTO 23
LN 0089      16 DO 17 I=1,10
LN 0090      II=I
LN 0091      IF(KAZON(1).EQ.INUM(IL))GOTO 18
LN 0092      17 CONTINUE
LN 0093      GOTO 21
LN 0094      18 J=2
LN 0095      GOTO 23
LN 0096      END

USASI-FORTRAN DIAGNOSTIC RESULTS FOR TAN21

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
BLOCK1      BLOCK2      BLOCK3      BLOCK4
BLOCK7      BLOCK9

```


USASI FORTRAN(2,0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001      SUBROUTINE  OUT 1
LN 0002      COMMON/BLOCK1/TR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP,/BLOCK2/
LN 0003      F10S7IM(14,0),IPIF(R),TAZPIF(2,50),IUTOMB(30),IUPUF(9),IPPR(1000),
LN 0004      MISCNP(50)
LN 0005      COMMON/BLOCK4/ISPACE,ISP,IS7AM(16),IFORM(13)
LN 0006      COMMON/BLOCK6/KUT,TAZI,IUH,KSSN(2),IFPUF(9)
LN 0007      COMMON/BLOCK8/IC
LN 0008      ICF1=1
LN 0009      ITYP=JAKP(TR,IRF1,L)
LN 0010      IF(ITYP,FO,4)GOTO 10
LN 0011      ICF2=2
LN 0012      J=JAKP(TR,TRE2,I)
LN 0013      IF(J,FO,0) RETURN
LN 0014      J=2*J+1
LN 0015      DO1 N=3,J,2
LN 0016      IM=N
LN 0017      DO3 I=1,9
LN 0018      3 IFPUF(I)=ISPACE
LN 0019      ISTD=JAKP(TR,IM,L)
LN 0020      IF(ISTD,EO,0) GOTO 1
LN 0021      K=IM+1
LN 0022      IXY=JAKP(IP,K,L)
LN 0023      ICF3=1
LN 0024      I=MOVFS(IFPUF,ICF3,ISTD)
LN 0025      ICFK=I-1
LN 0026      CALL TCHRS
LN 0027      TVRS=IOSZT(IY)
LN 0028      DO 1 IN=1,IREK
LN 0029      M=TVRS-1+IN
LN 0030      IOSZIM(IX,M)=IFPUF(IN)
LN 0031      1 CONTINUE
LN 0032      RETURN
LN 0033      10 ICF1=1
LN 0034      IHS7=1
LN 0035      CALL MCTAK(IST,IHSZ)
LN 0036      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR OUT1

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1	BLOCK2	BLOCK4	BLOCK6
BLOCK8			

```

USASI FORTRAN(2.0)/MASTER  INTEGER WORD SIZE = 2, * OPTION IS OFF, 0 OPTION IS OFF 09/04/72 PAGE 001

LN 0001      SUBROUTINE  ISTAT
LN 0002      COMMON/BLOK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP,/BLOK2/
LN 0003      FINS7IM(16,9),IPUF(R1,IA7PUF(2,50),IUTOMB(130),IUPUF(9),IPPR(1000),
LN 0004      MISNPP(50),/BLOK6/KUT,IAZI,IUH,KSSN(2),IFPUF(9)
LN 0005      COMMON/BLOK8/IC
LN 0006      ITYP=JAKP(IR,1,1)
LN 0007      IF(ITYP.NF.4)GOTO 10
LN 0008      M=JAKP(IR,7,L)
LN 0009      DO1 I=1,M
LN 0010      ISTD=JAKP(IR,2+I,L)
LN 0011      IF(ISTD.F0.0) GOTO 1
LN 0012      IF(ISTD.NF.9999)GOTO 2
LN 0013      NP=IOSZY(ISTD)
LN 0014      IF(NIP.F0.62)GOTO 6
LN 0015      DO3 K=1,NP
LN 0016      IK=KUT-1+K
LN 0017      3 IUTOMB(IK)=IPUF(K)
LN 0018      KUT=KUT+NP
LN 0019      1 CONTINUE
LN 0020      GOTO 8
LN 0021      2 J=MOVES(IUTOMB,KUT,ISTD)
LN 0022      GOTO 1
LN 0023      6 NDM=NP+1
LN 0024      DO7 K=2,NDM
LN 0025      7 IUPUF(K)=IPUF(K-1)
LN 0026      GOTO 8
LN 0027      10 CALL WTSTAK(1,4)
LN 0028      8 CONTINUE
LN 0029      RETURN
LN 0030      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR ISTAT

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
BLOK1      BLOK2      BLOK6      BLOK8

```


USAST FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 * * OPTION IS OFF * 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001 SUBROUTINE IUTLK
LN 0002 COMMON/BLK1/IR,IXY,IX,TY,TCXY,IAN,NIP,INP,ITOV,IP
LN 0003 COMMON/BLK2/IDSZIM(16,9),IPUF(8),IAZPUF(2,50),IUTOMR(30),
LN 0004 ITHPUF(9),IPPR(1000),ISNPP(50)
LN 0005 COMMON/BLK3/IFI,AG1,IFLAG2,IFLAG3,IFLAG4,IFLAG5,IFLAG6,IFLAG7
LN 0006 COMMON/BLK6/KUT,IAZI,IUM,KSSN(2),IFUP(9)
LN 0007 COMMON/BLK9/ IASSZ,IUSSZ,IAZON,INAM(2)
LN 0008 CALL NULL
LN 0009 IF (IFI,AG3,FQ,1) GO TO 2
LN 0010 IF (IX,FQ,6) GO TO 3
LN 0011 IF (IX,FQ,5) GO TO 4
LN 0012 IUSSZ=KONV1(IPUF)
LN 0013 IUSSZ=IUSSZ-1
LN 0014 4 DO 5 I=1,12
LN 0015 IUSSZ=IUSSZ+1
LN 0016 IF (IUSSZ,FQ,INP)GOTO 6
LN 0017 IUM=ISNPP(IUSSZ)
LN 0018 IUM0=IPPR(IUM)
LN 0019 IDSZIM(I,1)=IUSSZ
LN 0020 DO 5 K=1,IUM0
LN 0021 L=IUM+K
LN 0022 IDSZIM(I,K+1)=IPPR(L)
LN 0023 5 CONTINUE
LN 0024 7 IFLAG3=1
LN 0025 RETURN
LN 0026 2 IF (IFLAG4,FQ,0)GO TO 4
LN 0027 31 DO 9 I=1, 12
LN 0028 IUSSZ=IUSSZ-1
LN 0029 IF (IUSSZ,FQ,0) GO TO 6
LN 0030 IUM=ISNPP(IUSSZ)
LN 0031 IUM0=IPPR(IUM)
LN 0032 IDSZIM(I,1) =IUSSZ
LN 0033 DO 9 K=1,IUM0
LN 0034 L=IUM+K
LN 0035 IDSZIM(I,K+1)=IPPR(L)
LN 0036 9 CONTINUE
LN 0037 GO TO 7
LN 0038 3 IFLAG4 =1
LN 0039 IUSSZ=IUP
LN 0040 GO TO 31
LN 0041 6 I=40
LN 0042 CALL WIRAKT(L)
LN 0043 GO TO 7
LN 0044 END

```

USAST FORTRAN DIAGNOSTIC RESULTS FOR IUTLK

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLK1	BLK2	BLK3	BLK6
BLK5			

USAFI FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 * OPTION IS OFF * 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001      SUBROUTINE IPPUK
LN 0002      COMMON/BLOCK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003      COMMON/BLOCK2/IDSZ,IM(14,0),IPUF(8),IAZPUF(2,50),IUTOMR(30),
LN 0004      IIPUF(0),IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLOCK6/KUT,IAZI,TUM,KSSN(2),IPPUF(9)
LN 0006      COMMON/BLOCK9/ IASSZ,IUSSZ,IAZON,INAM(2)
LN 0007      IF(IX.EQ.0) GOTO 1
LN 0008      DO3 I=1, IAN
LN 0009      II=I
LN 0010      IF(IAZON.EQ.0,IAZPUF(2,II)) GOTO 4
LN 0011      3 CONTINUE
LN 0012      4 IAN=IAZPUF(1,I)
LN 0013      5 IIM=ISNPP(IAN)
LN 0014      IIMH=IPPR(IIM)
LN 0015      DO2 J=1,IUMH
LN 0016      K=IUMH+1
LN 0017      JK=J
LN 0018      2 IDSZ(M(1,5,IK))=IPPR(K)
LN 0019      RETURN
LN 0020      1 IAN=KONV1(IPUF)
LN 0021      IF(IAN.GE.INP) GOTO 7
LN 0022      GOTO 5
LN 0023      7 I=70
LN 0024      CALL HIRAKI(I)
LN 0025      RETURN
LN 0026      END

```

USAFI FORTRAN DIAGNOSTIC RESULTS FOR IPPUK

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1	BLOCK2	BLOCK6	BLOCK9
--------	--------	--------	--------


```
HEAST.FORTNANT2.01/WASTED  INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF  09/04/72  PAGE 001
```

```

LN 0001      SHROUTINE TUTORAL
LN 0002      COMMON/BLCK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003      COMMON/BLCK2/IDSZIM(I6,0),IPUF(R),IAZPUF(I2,50),IUTOMB(30),
LN 0004      IIPUF(9),IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLCK6/KUT,IAZT,IHH,KSSN(2),IFPUF(9)
LN 0006      IF (IX,FQ=5) GOTO 2
LN 0007      IF (IX,FQ=6) GOTO 1
LN 0008      ITORL2=KONV1(IIPUF)
LN 0009      3 IF (ITORL2.GF,INP) GOTO 4
LN 0010      ITORL=ITORL2-ITORL1+
LN 0011      ISS7=INP-(ITORL2+1)
LN 0012      K=ITORL2
LN 0013      DO5 I=ITORL1+ISSZ
LN 0014      K=K+1
LN 0015      5 ISNPP(I)=ISNPP(K)
LN 0016      L=INP-ITORL
LN 0017      N=INP-1
LN 0018      DO6 I=L,N
LN 0019      6 ISNPP(I)=0
LN 0020      INP=INP-ITORL
LN 0021      DO7 K=ITORL1,ITORL2
LN 0022      IA=IAN-1
LN 0023      DO7 I=1,IA
LN 0024      IL=I
LN 0025      IF (IAZPUF(I,IL),FQ,K) GOTO 8
LN 0026      GOTO 7
LN 0027      8 IAZPUF(I,I)=0
LN 0028      IAZPUF(2,I)=0
LN 0029      IP=I+1
LN 0030      DO7 M=IP,IA
LN 0031      IT=M
LN 0032      IAZPUF(I,IT)=IAZPUF(I,IL)+IDTORL
LN 0033      7 CONTINUE
LN 0034      RETURN
LN 0035      1 ITORL=KONV1(IIPUF)
LN 0036      RETURN
LN 0037      2 ITORL=KONV1(IPUF)
LN 0038      ITORL2=ITORL1
LN 0039      GOTO 3
LN 0040      4 L=70
LN 0041      CALL HIRAKI(I)
LN 0042      RETURN
LN 0043      END

```

USASI FORTRAN DIAGNOSTIC RESULTS FOR TUTOR1

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOK1	BLOK2	BLOK6
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27
28	29	30
31	32	33
34	35	36
37	38	39
40	41	42
43	44	45
46	47	48
49	50	51
52	53	54
55	56	57
58	59	60
61	62	63
64	65	66
67	68	69
70	71	72
73	74	75
76	77	78
79	80	81
82	83	84
85	86	87
88	89	90
91	92	93
94	95	96
97	98	99
100	101	102
103	104	105
106	107	108
109	110	111
112	113	114
115	116	117
118	119	120
121	122	123
124	125	126
127	128	129
130	131	132
133	134	135
136	137	138
139	140	141
142	143	144
145	146	147
148	149	150
151	152	153
154	155	156
157	158	159
160	161	162
163	164	165
166	167	168
169	170	171
172	173	174
175	176	177
178	179	180
181	182	183
184	185	186
187	188	189
190	191	192
193	194	195
196	197	198
199	200	201
202	203	204
205	206	207
208	209	210
211	212	213
214	215	216
217	218	219
220	221	222
223	224	225
226	227	228
229	230	231
232	233	234
235	236	237
238	239	240
241	242	243
244	245	246
247	248	249
250	251	252
253	254	255
256	257	258
259	260	261
262	263	264
265	266	267
268	269	270
271	272	273
274	275	276
277	278	279
280	281	282
283	284	285
286	287	288
289	290	291
292	293	294
295	296	297
298	299	300
301	302	303
304	305	306
307	308	309
310	311	312
313	314	315
316	317	318
319	320	321
322	323	324
325	326	327
328	329	330
331	332	333
334	335	336
337	338	339
340	341	342
343	344	345
346	347	348
349	350	351
352	353	354
355	356	357
358	359	360
361	362	363
364	365	366
3		


```

USASI FORTRAN(2.0)/MASTER    INTEGER WORD SIZE = 2 * * OPTION IS OFF * 0 OPTION IS OFF    09/04/72    PAGE 001

LN 0001      SUBROUTINE IUREH
LN 0002      COMMON/BLOCK1/IR,IXY,IX,IY,ICXY,IAN,NIP,INP,ITOV,IP
LN 0003      COMMON/BLOCK2/IDCZIN(16,9),IPUF(8),IAZPUF(2,50),IUTOMB(30),IUPUF(9)
LN 0004      1,IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLOCK3/IFLAG1,IFLAG2,IFLAG3,IFLAG4,IFLAG5,IFLAG6,IFLAG7
LN 0006      IF(IX.EQ.3)GOTO 1
LN 0007      IUS2=CONV1(IPUF)
LN 0008      IF(IUS2.GE.INP)GOTO 2
LN 0009      DO 3 I=1,INP
LN 0010      K=INP-I
LN 0011      IF(K.LT.IUS1)GOTO 4
LN 0012      3 ISNPP(K+1)=ISNPP(K)
LN 0013      4 ISNPP(IUS1)=ISNPP(IUS2+1)
LN 0014      DO5 I=IUS2,INP
LN 0015      5 ISNPP(I)=ISNPP(I+1)
LN 0016      DO6 K=IUS1,IUS2
LN 0017      I=IUS1
LN 0018      DO6 I=1,IA
LN 0019      IF(IAZPUF(I,I).EQ.K)GOTO 7
LN 0020      GOTO 6
LN 0021      7 IF(K.EQ.IUS2)GOTO 8
LN 0022      IAZPUF(I,I)=IAZPUF(I,I)+1
LN 0023      6 CONTINUE
LN 0024      8 IAZPUF(I,I)=IUS1
LN 0025      RETURN
LN 0026      2 L=70
LN 0027      CALL WIRAKT(L)
LN 0028      RETURN
LN 0029      1 IUS1=CONV1(IPUF)
LN 0030      RETURN
LN 0031      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR IUREH

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1    BLOCK2    BLOCK3

```

USAST FORTRAN(2,0)/MASTED INTEGER WORD SIZE = 2 * * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 003

```

LN 0001      SUBROUTINE KOSIKI
LN 0002      COMMON/BLOCK1/IP,IXY,IX,TY,ICXY,IAN,NIP,INP,IIOV,IP
LN 0003      COMMON/BLOCK2/IOSZIM(16,0),IPUF(8),IAZPUF(2,50),IUTOMB(30),IUPUF(9)
LN 0004      1,IPPR(1000),ISNPP(50)
LN 0005      COMMON/BLOCK6/RUT,IAZI,IUH,KSSN(2),IFPUF(9)
LN 0006      CALL NULL
LN 0007      IF(IX,EO,12) GO TO 1
LN 0008      I=2
LN 0009      4 ISSZ=KSSN(I)
LN 0010      IF(ISSZ,EO,0) GO TO 2
LN 0011      IUH=ISNPP(ISSZ)
LN 0012      ID8=IPPR(IUH)
LN 0013      DO3 I=1,ID8
LN 0014      K=IUH+I
LN 0015      IOSZIM(5,I)=IPPR(K)
LN 0016      3 CONTINUE
LN 0017      RETURN
LN 0018      1 I=1
LN 0019      RATO 4
LN 0020      2 IF(I,EO,1) GO TO 4
LN 0021      6 L=72
LN 0022      CALL MTRAKT(L)
LN 0023      RETURN
LN 0024      5 L=71
LN 0025      GOTO 4
LN 0026      END

```

USAST FORTRAN DIAGNOSTIC RESULTS FOR KOSIKI

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

BLOCK1 BLOCK2 BLOCK6


```

USASI-FORTRAN(2.0)/MASTER  INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF  09/04/72  PAGE 001

LN 0001      FUNCTION KONV1(TPUF)
LN 0002      COMMON/BLOCK 4/ TSPACE, ISP, ISZAM(10), IFORM(3)
LN 0003      COMMON/BLOCK12/IFR, TPUF(2)
LN 0004      CHARACTER KPUF(10), KER(2), ISP
LN 0005      EQUIVALENCE (KPUF, TSPUF), (KER, IFR)
LN 0006      TPUF(1)=TPUF(1)
LN 0007      TPUF(2)=TPUF(2)
LN 0008      IFLAG=0
LN 0009      TSP=TSPEC
LN 0010      DO1 I=1, 10
LN 0011      IF (KPUF(I).EQ.TSP) GOTO 2
LN 0012      IFLAG=1
LN 0013      KER(I)=KER(I)*10+KPUF(I)
LN 0014      KONV1=IFR
LN 0015      1 CONTINUE
LN 0016      RETURN
LN 0017      2 IF (IFLAG.EQ.1) RETURN
LN 0018      GOTO 1
LN 0019      END

      USASI FORTRAN DIAGNOSTIC RESULTS FOR KONV1

      NO ERRORS

      THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

      BLOCK4      BLOCK12

```


USASI FORTRAN(2.0)/MASTER INTERPRETER SIZE = 2 * * OPTION IS OFF + 0 OPTION IS OFF 09/04/72 PAGE 001

LN 0001	SUBROUTINE MIRAKI(1)
LN 0002	DIMENSION IFPUF(9)
LN 0003	COMMON/ALOK2/IDSZIM(16,9),IPUF(8),IAZPUF(2*50),IUTOMB(30),IUPUF(9)
LN 0004	F,IPR(1000),ISNPP(50)
LN 0005	K1
LN 0006	J=MOVES(IFPUF,K,1)
LN 0007	DOI 1=1,9
LN 0008	1 IDSZIM(6,1)=IFPUF(1)
LN 0009	RETURN
LN 0010	END

USASI FORTRAN DIAGNOSTIC RESULTS FOR MIRAKI

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

ALOK2

USAST FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001        FUNCTION ISURR(IR)
LN 0002        ITYP = JAKP(IR,1,L)
LN 0003        IF(ITYP.EQ.0) GOTO 1
LN 0004        TSURR=TD
LN 0005        RETURN
LN 0006        1 DO2 T=1,L+3
LN 0007        T1=T
LN 0008        M1=JAKP(IR,T1,M)
LN 0009        M11=M1+1
LN 0010        M2=JAKP(IR,MU,M)
LN 0011        M11=M11+1
LN 0012        M2=JAKP(IR,MU,M)
LN 0013        CALL WAPF(M1,M2,M11)
LN 0014        2 CONTINUE
LN 0015        TSURR=JAKP(IR,2,M)
LN 0016        RETURN
LN 0017        END
```

USAST FORTRAN DIAGNOSTIC RESULTS FOR ISURR

NO ERRORS

USASI FORTRAN(2.4)/MATER INTERFER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001      SUBROUTINE WARF(IP,N,NEW)
LN 0002      COMMON/DIALTAB/ITAB(5000)
LN 0003      COMMON/DIALCO/IS,IA,ISV,IAV,IST,IFL(10),J,ILIM
LN 0004      JAV1=IAV*IP-1
LN 0005      ICA=ITAB(JAV1)+1
LN 0006      JAV2=ICA*N
LN 0007      ITAB(JAV2)=NEW
LN 0008      RETURN
LN 0009      END
```

USASI FORTRAN DIAGNOSTIC RESULTS FOR WARF

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

DIALTAB	DIALCO
---------	--------


```

USAST FORTRAN(2.0)/M=STEP      INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF      09/04/72      PAGE 001

LN 0001      FUNCTION MOVES(IPUF,I,TSP)
LN 0002      DIMENSION IPUF(10)
LN 0003      COMMON/DISCOF/IST(2*100),LAP(2*50),NUMW(50),INOP,IFIRST(2)
LN 0004      COMMON/DIALTA/ITAR( 5000)
LN 0005      COMMON/DIALCO/TS,TA,TSV,IAV,IST,IFL(10),J,ILIM
LN 0006      COMMON/SZOVEG/ISZ(17),TSE
LN 0007      MSIZE=TSV
LN 0008      IF(TSP)1,1,2
LN 0009      1 ISPP=TSF
LN 0010      DO 3 K=1,50
LN 0011      IF(TSP,IF,NUMW(K)) GOTO 4
LN 0012      3 ISPP=ISPP+NUMW(K)
LN 0013      STOP K1
LN 0014      4 IF(K.FQ,INOP) GOTO 6
LN 0015      RCHTND 4
LN 0016      MSIZE=LAP(2,K)
LN 0017      LIM=LAP(1,K)+1
LN 0018      IF(LIM.FQ,0) GOTO 8
LN 0019      DO 5 J=1,LIM
LN 0020      5 READ(4)
LN 0021      8 LIM=MSIZE+4000
LN 0022      LIM=LIM+1
LN 0023      LTMF=LIM+MSIZE
LN 0024      INOP=K
LN 0025      READ(4)ISZ,ITAR(K),K=LIM,LTMF
LN 0026      6 MSIZE=TSV
LN 0027      TSV=TSZ(3)
LN 0028      TSF=ILIM
LN 0029      GOTO 7
LN 0030      2 ISPP=TSF
LN 0031      TSF=0
LN 0032      7 MOVES=MOVES+IPUF,I+ISPP
LN 0033      TSV=MSIZE
LN 0034      RETURN
LN 0035      END

USAST FORTRAN DIAGNOSTIC RESULTS FOR MOVES

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
DISCOF      DIALTA      DIALCO      SZOVEG

```

```

USASI FORTRAN(2.01)/MASTED  INTEGER WORD SIZE = 2 ; * OPTION IS OFF ; 0 OPTION IS OFF  09/04/72  PAGE 001

LN 0001      FUNCTION MOVED(IPUF,I,ISZP)
LN 0002      DIMENSION IFL(10),IPUF(10)
LN 0003      COMMON/DIALCO/ YS,IA,ISV,IAV,ISI,IFL,J,ILIM
LN 0004      COMMON/SZOVEG/ISZ(17),ISE
LN 0005      COMMON/DIALTAB/ITAB(5000)
LN 0006      JAVI=[SV*ISZP-1      +ISE
LN 0007      KFX=ITAB(JAVI)      +ISE
LN 0008      NFX=ITAB(KFX)
LN 0009      LIM1=KFX+1
LN 0010      LIM2=KFX+(NFX+7)/8
LN 0011      DO 1 J=LIM1,LIM2
LN 0012      IPUF(I)=ITAB(J)
LN 0013      1 I=I+1
LN 0014      MOVED=I
LN 0015      RETURN
LN 0016      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR MOVED

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
DIALCO      SZOVEG      DIALTAB

```



```
USASI FORTRAN(2.0)/MASTER  INTEGER WORD SIZE = 2 , * OPTION IS OFF , * OPTION IS OFF  09/04/72  PAGE 001

LN 0001      FUNCTION ICSAT(K)
LN 0002      IND=JAKP(K,1,L)
LN 0003      IF (IND) 1+2+2
LN 0004      2  ICSAT=K
LN 0005      RETURN
LN 0006      1  ICSAT=JAKP(K,2+1)
LN 0007      IND=IND
LN 0008      TAIL=STO(IND)
LN 0009      RETURN
LN 0010      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR ICSAT

NO ERRORS
```



```

USAST FORTRAN12.01/MASTER    INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS  OFF    09/64/72    PAGE 001

LN 0001      FUNCTION IARF (IR,N,L)
LN 0002      DIMENSION IFL(10)
LN 0003      COMMON/DIAL TAB/TIARI 5000)
LN 0004      COMMON/DIAL CO/ IS,IA,ISV,IAV,IST,IFL,J,ILIM
LN 0005      JAV1=IAV+IR-1
LN 0006      ICA=ITAR(JAV1)-1
LN 0007      JAV2=ICA+N
LN 0008      IARF=ITAB(JAV2)
LN 0009      L=ITAR(ICA+1)*2
LN 0010      KRU=ICA+L+2
LN 0011      L=L+ITAR(KRU)
LN 0012      KDUK=1
LN 0013      LTM=ITAR(KRU)
LN 0014      DO 2 I=1,LTM
LN 0015      JAV3=KRU+KDUK
LN 0016      L=L+ITAR(JAV3)
LN 0017      2 KDUK=KDUK+ITAR(JAV3)
LN 0018      RETURN
LN 0019      END

USAST FORTRAN DIAGNOSTIC RESULTS FOR IARF

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

DIAL TAB    DIAL CO

```

```
USASI-FORTRAN(2.0)/MASTER    INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF    09/04/72    PAGE 001

LN 0001      FUNCTION JAKP(IR,N,L)
LN 0002      DIMENSION IFL(10)
LN 0003      COMMON/DIAL TAB/ITAB( 5000)
LN 0004      COMMON/DIAL CO/ IS,IA,ISV,IAV,IST,IFL,J,ILIM
LN 0005      JAV=IAV+IR-1
LN 0006      ICA=ITAB(JAV)
LN 0007      I=ITAB(ICA)
LN 0008      JAV2ICA=N
LN 0009      JAKP=ITAB(JAV2)
LN 0010      RETURN
LN 0011      END

USASI FORTRAN DIAGNOSTIC RESULTS FOR JAKP

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE
DIALTAB    DIALCO
```

USASI FORTRAN(2,0)/MASTER INTEGER WORD SIZE = 2 * OPTION IS OFF * 0 OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001      FUNCTION NER(IR)
LN 0002      DIMENSION IFL(10)
LN 0003      COMMON/DIALTAB/ITAB( 5000)
LN 0004      COMMON/DIALCO/ IS,IA,ISV,IAV,IST,IFL,J,ILIM
LN 0005      JAVI=IAV+ID-1
LN 0006      ICA=ITAB(JAVI)
LN 0007      ICAT=ICA+ITAB(ICA)*J
LN 0008      NER=ITAB(ICA)
LN 0009      RETURN
LN 0010      END
```

USASI FORTRAN DIAGNOSTIC RESULTS FOR NER

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

DIALTAB DIALCO

USAST-FORTRAN(2.0)/MASTER INTEGER WORD SIZE = 2 , * OPTION IS OFF , 0 OPTION IS OFF 09/04/72 PAGE 001

```

LN 0001      FUNCTION JFLP(IP,INT,N,I)
LN 0002      DIMENSION IFL(10)
LN 0003      COMMON/DIALTAB/ITAB( 5000)
LN 0004      COMMON/DIALCO/ IS,IA,ISV,IAV,IST,IFL,J,ILIM
LN 0005      JAV1=IAV*ID=1
LN 0006      ICA=ITAB(JAV1)
LN 0007      TCA1=ICA+ITAB(TCA1)+2
LN 0008      LIM1=INT-1
LN 0009      IF (LIM1.EQ.0) GO TO 3
LN 0010      DO 2 I=1,LIM1
LN 0011      2 TCA1=ICA1+ITAB(TCA1)+1
LN 0012      3 I=ITAB(TCA1)
LN 0013      JAV2=ICA1+N
LN 0014      JFLP=ITAB(JAV2)
LN 0015      RETURN
LN 0016      END

```

USAST-FORTRAN DIAGNOSTIC RESULTS FOR JFLP

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

DIALTAB DIALCO

USAST FORTRAN(2,4)/MASTER INTEGER WORD SIZE = 2, * OPTION IS OFF, 0 OPTION IS OFF 09/04/72 PAGE 001

```
LN 0001      FUNCTION INPAG(KEZD)
LN 0002      COMMON/DISCOP/ 1IST(2+100),LAP(2+50),NUMW(50),INOP,IFIRST(2)
LN 0003      INPAG=0
LN 0004      IF(KEZD1).1.2
LN 0005      2  MSIZE=STO(KEZD)
LN 0006      RETURN
LN 0007      1  INPAG=IFIRST(2)
LN 0008      MSIZE=STO(IFIRST(1))
LN 0009      RETURN
LN 0010      END
```

USAST FORTRAN DIAGNOSTIC RESULTS FOR INPAG

NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

DISCOP
RLDR(EXT,FAIR,LGO)

5. Függelék

AZ INTERAKTIV ALKATRÉSZPROGRAM IRÁSÁNAK FOLYAMATA
A SZIMULÁCIÓS AIR PROGRAM OUTPUTJA

Az 5. függelékben bemutatjuk az interaktív alkatrészprogram írásának folyamatát a CDC 3300-as számítógépen megvalósított szimulációs AIR program segítségével. A display-n megjelenő képeket szélesnyomtatón irattuk ki. A szimulációban a képernyőn történő minden változásnak egy-egy kirajzolt kép felel meg, amely lehetővé teszi az egymás utáni lépések követését. Az operátor válaszait lyukkártyákra lyukasztva adjuk meg, amelyeket a rendszer olvas be. A rámutatásnak megfelelő válasz megadását a cursor-cím beolvasásával valósítottuk meg.

A kirajzolt képek nagysága nyolc karakterrel több oszlopot tartalmaz, mint az eredeti képernyő. Ezt a számítógép kötött szóhosszusága tette szükségessé, hogy az egyes szövegek esetleges egymásra írását elkerüljük.

A közölt példák a 2. függelékben szereplő dialógusképek megvalósítását mutatják be.

PART NO

A7-AIKATPFS7 NFVE0

PARTNO

A7 ALKATRES7 NFVE0
1 EME7 721


```
*POST=PROCESSOR
```

MACHIN

VAN < >
NINGS < >

```

**MACHIN**
**< >**
**VAN NTNCS < > **
**POST-PROCESSOR**

```

00000002

POST-PROCESSOR

VAN NINGS

MACHIN

VISSZA < >

NOPOST

```
*00000002          *MACHIN  
*  
*POST-PROCESSOR    *      < >  
*VAN                *      ↑  
*NTNCS              *
```

VISSA < >

<

```
*NOPOST
```


二二

001 TAPE-ROL I I STAT KEREK
002 NEW KEREK

✓ ✓

*CI TAPF=ROL I ISTAT KFERFK
* NEM KFERFK

CLARNT

< > VTSSZA

C1 PRINT

$\begin{matrix} \nearrow & \rightarrow \\ \downarrow & \end{matrix}$

V7556A

TRANS

KATRESZ KOORDINATARENDSZER ORIGOJA
##4 SZERSZAMGEP ASZTALON

07 04 00

▲ ▲ ▲

TRANS

#AL KATRESZ KOORDINATAKENDSZER ORIGOJA
##A SZERSZANGEP ASZTALON

XO YO ZO

521

TRANS

AL KATRÉSZ KOORDINÁTARENDSZER ORIGÓJA
A SZERSZÁMGÉP ASZTALON

X0	<	125
Y0	<	125.5
Z0	<	

VISSZA < >

TRANS

AL KATRÉSZ KOORDINÁTARENDSZER ORIGÓJA
A SZERSZÁMGÉP ASZTALON

X0	<	125
Y0	<	125.5
Z0	<	150

VISSZA < >

TRANS.

00000004

#AJKATRSZ KOORDINATARENDSEZ ORIGOJA

AA-S7EPS7AMGCP ASZTAION

X0	✓	✓	125
----	---	---	-----

125.0

150

V7551A

#TRANS/125.125.5.150

FRANIS

0000000000

#AIVATPFC7 K000DINA TAREND057FR ORTGO-IA

57FBS7AMGSP AS7YAL ON

125

Y0	<	125.0
----	---	-------

20 150

A7581A ()

0913561-195 5.150

五九四

WYKATRSZ ANYGANAK SZANA

PART

41 KATRESZ ANYAGANAK SZÁMA

91

PART	
16	
1. ELOMUNKALATLANOK	< >
2. ELOMUNKALTAK	< >
3. ELOONTOTTEK	< >
VISSZA	< >

AL KATRESZ ANYAGANAK SZAMA	16	PART
MEGUNKALAST HELYEK	1. ELOMUNKALATLANOK	< >
	2. ELOMUNKALTAK	<
	3. ELOONTOTTEK	< >
HA KERDESRE NEM KOTELEZO VALASZOLNI	VISSZA	< >

PART	
ALKATRÉSZ ANYAGÁNAK SZÁMA	16
MEGMUNKALÁST HELYEK	
1. ELOMUNKALATLANOK	< >
2. ELOMUNKALTAK	↑
3. ELOONTOTTEK	< >
MUNKADARAB FELÜLETE	
1. MEGMUNKALT	< >
2. NYERS	< >
KÉRDESRE NEM KOTFELÉZO VALASZOLTI	VISSZA < >

PART	
ALKATRÉSZ ANYAGJAINAK SZÁMA	16
MEGMUNKALÁST HELYEK	< >
1. ELOMUNKALATLANOK	
2. ELOMUNKALIAK	↑
3. ELOONTOTTAK	< >
MUNKADARAB FELÜLETE	< >
1. MEGMUNKALT	
2. NYERS	Σ
HA KÉRDESHÉ NEM KÖTELEZŐ VÁLASZOLNI	VISSZA < >

PART	
AI KATHESZ ANYAGNAK SZAMA	16
MEGMUNKALAST HELYEK	< >
1. ELOMUNKALATANOK	<
2. ELOMUNKALITAK	<
3. ELOONTOTTEK	< >
MUNKADARAB FELULET	< >
2. NYERS	<
ORGACSIKAST PARAMETEREK MODOSITASA	<
1. ELOTOLAS	<
2. FORDULATSZAM	<
A KERDESRE NEVE KOTELEZO VALASZOLNI	VISSZA < >

* * * * *	000000005		PART	*
* * * * *				*
* * * * *	**ALKATRESZ ANYAGANAK SZAMA		16	*
* * * * *			"	*
* * * * *	**MEGMUNKALAST HELYEK		"	*
* * * * *			< >	*
* * * * *	1. ELOMUNKALATANOK		<	*
* * * * *	2. ELOMUNKALIATAK		<	*
* * * * *	3. FLOONTOTTEK		< >	*
* * * * *				*
* * * * *	**MUNKADARAB FELULET		< >	*
* * * * *			<	*
* * * * *	1. MEGMUNKALT		<	*
* * * * *	2. NYERS		<	*
* * * * *	**FORGACSOLAST PARAMETEREK MODOSITASA		<	*
* * * * *			<	*
* * * * *	1. ELOTOLAS		<	*
* * * * *	2. FORDULATSZAM		<	*
* * * * *			<	*
* * * * *	**A KERDESRE NEM KOTFELEZO VALASZOLNT		VTSSZA < >	*
* * * * *	**PART/MATERL.16,SEMI,ROUGH			*

*****00000005*****		*****PART*****	
ALKATRÉSZ ANYAGNÁK SZÁMA	16		
MEGMUNKALÁST HELYEK		<	>
1. ELOMUNKALATLANOK			
2. ELOMUNKALITAK			
3. ELOONTOTIEK		<	>
MUNKADARAB FELLETE		<	>
2. NYERS			
FORGACSOLÁSI PARAMETEREK MODOSITÁSA			
1. ELOTOLAS		<	>
2. FORDULATSZÁM		<	>
AA KÉRDESRÉ NEM KÖTELEZŐ VÁLASZOLNI		VISSZA	<
APART/MATERL, 16, SFMT, ROUGH			<
*****SAFPOS*****			
SZERSZÁMCSERÉ HELYE		EGY PONTBAN	<
		SIKHAN	<


```
*****  
* SAFPOS *  
*****  
* EGY PONTBAN < >  
* SIKHAN ^  
*****  
* SIK Z KOORDINATAJAO  
*****  
* VISSZA < >  
*****
```

* SAFPOS

* SZERSZAMCSERE HELYE
* EGY PONTBAN < >
* SIKHAN ↑

* STK Z KOORDINATAJAO
* 125

* VISSZA < >

* SAFPOS

* SZERSZAMCSERE HELYE
* EGY PONTBAN < >
* SIKHAN ↑

* STK Z KOORDINATAJAO
* 125

* VISSZA < >

* SAFPOS/125

0000000000

MS7FRSZAMCSFDE HFLYF

EGY PONTBAN
SIKHAN

851K Z KORDYNATA JAO

125

VTSZA < >

521/5003125

MMF1YIK FFJF7FTFT KFRFD

- #1. GEOMETRIAI MEGHATÁROZÁSOK
- #2. TECHNOLOGIAI MEGHATÁROZÁSOK
- #3. LEHETŐSÉK, VEGREHÁJTÁSOK
- #4. EGYEB PROGRAMTECHNIKAI UTASÍTÁSOK
- #5. PROGRAMMANTUPULATOK
- #6. ALKATRÉSZPROGRAM VEGE

*MELYIK FEJEZETET KEREDA	

*MEGHATÁROZÁSOK	
1. SÍK	< >
2. EGYEDI PONT	<
3. EGYENES	< >
4. KÖR	< >
5. PONTMINTAZAT	< >
6. KOORDINÁTARENSZER TRANSZFORMÁCIÓ	< >

VISSZA < >	

*MELYIK FEJEZETET KEREDA	

*MEGHATÁROZÁSOK	
1. SÍK	< >
2. EGYEDI PONT	<
3. EGYENES	< >
4. KÖR	< >
5. PONTMINTAZAT	< >
6. KOORDINÁTARENSZER TRANSZFORMÁCIÓ	< >

VISSZA < >	

*2. EGYEDI PONT	

POINT

A PONT ELHELYEZKEDIKO

1. SIKBAN

2. TERREN

3. PONTMINTAZAT EGY ELEME

VISSZA < >

POINT

A PONT ELHELYEZKEDIKO

1. SIKBAN

2. TERREN

3. PONTMINTAZAT EGY ELEME

VISSZA < >

2. TERREN

VYSSLA < >

POINT

A POINT AZONOSTOJAO

* * 02

VISSA \$ >

[illegible]

[illegible]

```
*****  
**          POINT  
  
*****  
**              *      p2  
  
*****  
**                  X0   < -    36  
**                  Y0   < -    36.5  
**                  Z0   <     ^  
  
*****  
**                VISSZA < >  
  
*****
```



```
**00000009**POINT**  
*  
*  
*  
**A POINT AZONOSITOJAO P2  
  
*  
*COORDINATAIO  
*  
*XO < - 36  
*YO < - 36.5  
*ZO < - 3  
*  
*  
*  
*  
*  
*  
*VISSZA < >  
*  
*P2=POINT/36.36.5.3
```


#GFCMF TOTAL PES7801 KTI FPEK0

SMAS GEOMETRIAI FEJFZETET KERÉKO

AF 10254 IVAS

BUJAH PONTOT AKAHOK MEGHATARONIO

RENAS GEOMETRIAT FEJFZETET KERÉKO

ВЕРХНЕУРАЛЬСКИЙ РАЙОН

VTSS/A < >

THOMAS GEOMFIRIAT FF JFZFIFY KFRFKO

FFJF7FFET KERENA

0050VZUBVTAHGFN40

1. 5TK

2. EGYPTI PONT

3. EGYPT S

4. Кор

5. PONTINATAZAT

6. KOORDINATARENDSZER TRANSZFORMACIO

< 07551A >

FFLYIK FFJFZETET KEREDA	
FFGHATAROZASOK	
1. SIK	< >
2. EGYEDI PONT	< >
3. EGYENES	<
4. KOR	< >
5. PONTMINTAZAT	< >
6. KOORDINATARENSZER TRANSZFORMACIO	< >
VISSZA	< >

IRODALOMJEGYZÉK

- [1] ALLAN, J.J. Fundamentation of the Many Manifestations of Computer Augmented Design
IFIP Working Conference on Principles of CAD,
Eindhoven, 1972. okt. 16-18.
- [2] AMBROZY, D. Dialogus, dialogusrendszerek.
MTA SzTAKI Közlemények 1971.
- [3] AMBROZY, D. On Man Machine Dialogue
Int. J. Man-Machine Studies. 3, 3
- [4] ANON, Computer Control Can be Cheap
Control Engineering, 15, No. 1. 1968. jan.
- [5] ANON, GE Makes an Entry Into Computerized NC...
Control Engineering, 16, No. 7. 1969. jul.
- [6] BARANSZKY-JOB I. Automatikus szerelés a gépiparban
Műszaki Gazdasági Tájékoztató 1970/3.
- [7] BARKER, A.J. Operating Experience with a Molins Machine
Machinery and Production Engineering 1969. nov.26.
- [8] BARKER, A.J. Some Recent Machine Tool Developments in
Russia and Other East European Countries
Machinery and Production Engineering 1969.nov.12.
- [9] BARKER, A.J. "Probomat" Numerical Control System for
Machine Tools
Machinery /GB/ 104, No. 2682. 814/1964.
- [10] BLEE, M. Interactive Planning
Data Systems 1970. aug.
- [11] BORKO, H.-BURNAUGH, H.P. Interactive Displays for
Document Retrieval
Information Display 1966. szept/okt.

- [12] BRANKAMP, K.-JUNGHANS W.-BERGER H. Wirtschaftlicher Einsatz von Bearbeitungszentren
Industrie Anzeiger No. 95. 1969. nov. 14.
- [13] BROSHEER, B.C. New DNC System Unveiled
American Machinist 1970. aug. 24.
- [14] BUDZILOVICH, P.N. Computerized NC-A Step Toward the Automated Factory
Control Engineering No. 7. 1969. jul.
- [15] BURGON, C.J. Computer Controlled Machine Tools
Metalworking Production 1970. márc. 25.
- [16] BURTON, H. Which Way for NC Tape Preparation?
Metalworking Production, 1969. nov. 12.
- [17] BUTLIN, G.A.-HUBBOLD, R.J. A Scheme for Man-Machine Interactive Structural Analysis
International Conference on Computer Aided Design The University of Southampton 1969.
ápr. 15-18.
- [18] CARLSON, R.D. Taking a Plunge in DNC
American Machinist 1969. jul. 28.
- [19] CHASEN, S.H. The Introduction of Man-Computer Graphics into the Aerospace Industry
Proc. Fall Joint Comp. Conf. 1965. p 883-892.
- [20] CHASEN, S.H.-SEITZ, R.N. On-Line Systems and Man-Computer Graphics
Computer and Automation, 1967. nov.
- [21] CHINGARI, G. The Numerical Control Information Utility: Concepts and Considerations
Proceedings A.C.N. National Meeting, 1967.

- [22] CLAUSSES, U. Methodical Design the Bases of Computer Aided Design
IFIP Working Conference on Principles of CAD
Eindhoven 1973. okt. 16-18.
- [23] CLAY, L.D.-LONG, F.J. Transfer Line "Memorizes" Job.
American Machinist, 1969. szept. 22.
- [24] COONS, S.A. Computer Graphics and Innovative engineering design
Datamation, 1966. máj.
- [25] COONS S.A. The Uses of Computers in Technology
Scientific American, 215.3 1966. márc.
- [26] JANTZEN, K. The Functions of a postprocessor
Metalworking Production, 1968. márc. 13.
- [27] CREMERIUS, J. APT/IGS State of the Art in NC Graphics
Prolamat'73 Conference 1973. ápr. 10-13.
- [28] CRESTIN, J.P.-PAILLARD J.F. A Small Graphical System
for Programming NC Machines
Prolamat'73 Conference 1973. ápr. 10-13.
- [29] CZENTE, SZ. Számirányításu /NC/ szerszámgépek kézi
programozása
Gépipari Tudományos Egyesülés Közlemények
84/1971.
- [30] DAVIES, K.J. GNC-A Graphical NC Processor
Prolamat'73 Conference 1973. ápr. 10-13.
- [31] DAVIES, K.J. Interaktív számítógépes grafika-rendszer
és alkalmazása a gépipari tervezésben
CAD szeminárium (CAD Centre-MTA SztAKI) Budapest
1973. ápr. 16-18.
- [32] DRAYTON, C.E.-GRAY, H. Automatically programmed Tools
Prolamat Conference Roma, 1969. szept. 15-17.

- [33] ENDRŐDI T., "SZÁGA" (Számítógépes adatelőkészítés) .
MTA SzTAKI Tanulmány 1969.
- [34] FEINBERG, B. Computer Boosts Transfer-Line Efficiency
The Tool and Manufacturing Engineer 1969. okt.
- [35] FEINBERG, B. Computer Assisted Programming for NC
Machine Tools
The Tool and Manufacturing Engineer 1969. szept.
- [36] FENICHEL, R. A Program to Teach Programming
Communications of the ACM 1970. márc.
- [37] FERENCZY, J. An Examination of Alternative Methods of
Strip Layout Design for Press Tools Using a
Computer
CAD Centre Report No. 218/72 1972. ápr.
- [38] FERENCZY, J.-KOVÁCS, M. Számítógépes interaktív
tervezés
MTA SzTAKI Közlemények 5. sz. 1969.
- [39] FORGÁCS, T.-HERMAN, G.-PIKLER, G. Software Methods in
Developing CAD Programs
IFIP Working Conference on Principles of CAD
Eindhoven 1972. okt. 16-18.
- [40] FORGÁCS, T.-GERHARD, G.-KOC SIS, J.-KRAMMER, G. DISTAR-B
általános dialógus rendszer
MTA SzTAKI "Felhasználói dokumentáció" 1971. okt.
- [41] GAUDISSERT, P. Problems Arising When Implementing Point-
to Point Programming Languages. An Approach to
a Solution
Prolamat Conference Roma, 1969. szept. 15-17.
- [42] GENTHNER, H.J. Interactive Computer Graphics-Shrinking
the Decision-Making Loops
Design News 1968. szept. 16.

- [43] GIDDINGS, F.A. Computer-Assisted N/C Programming Automation, 1969. okt.
- [44] GARRATT, G. NC Applied to Large Batch Production Machinery and Production Engineering 1970. febr. 25.
- [45] GOTT, B. Egy nemzeti számítógépes tervezési központ CAD szeminárium (CAD Centre-MTA SzTAKI) Budapest 1973. ápr. 16-18.
- [46] GOTT, B. The Scope of Computer-Aided-Design IFIP Working Conference on Principles of CAD Eindhoven 1972. okt. 16-18.
- [47] GRALAK, A.-GRALAK, W. Systemy przygotowania informacji do szterewania liczbowego obrabiarek Mechanik 5. sz. 1969. máj.
- [48] GREEN D.-CORNISH R. Displaytran a Graphic Oriented Conversational System Proceeding-Full Joint Computer Conf. 1969.
- [49] GROAT, D.G. Programming in "SHOP" American Machinist 1968. jan. 29.
- [50] GYÜRKI, J. Tanulmány az adaptív szerszámgép-irányítás /AC/ területén MTA SzTAKI Tanulmány 3. 1972. márc.
- [51] HALÁSZ, S.-MÁRKUS, T. Pályavezérlésű szerszámgépek programnyelveinek honosítása "Compcontrol'70" Konferencia Miskolc 1970. jul. 7-11.
- [52] HATSCHEK, R.L. Computers Take Control American Machinist 1970. aug. 10.
- [53] HATSCHEK, R.L. NCS Spotlights Computers American Machinist 1970. máj. 4.

- [54] HATVANY, J. Számvezérlésű szerszámgépek programozása -
számítógépes eljárások
MTA SzTAKI Közlemények 2. sz. 1969.
- [55] HATVANY, J. EXAPT2, az NC esztergák szimbolikus prog-
ramnyelve
MTA SzTAKI Közlemények 2. sz. 1969.
- [56] HATVANY, J.-KRAMMER, G. Do Graphic Languages Need to be
Problem-Oriented?
"Graphic Languages" Ed. F. Nake and A. Rosenfeld
North-Holland, Amsterdam 1972.
- [57] HATVANY, J. Új módszer a számjegyes szerszámgépvezérlés
alkalmazásában
OMFB ismertető tanulmány 4-811-55 1968. okt.
- [58] HATVANY, J. The Engineer's Creative Activity in a CAD
environment
IFIP Working Conference on Principles of CAD,
Eindhoven 1972. okt. 16-18.
- [59] HEIDEMAN, M.V. Integrated Manufacturing Cycle with
Numerical-Control Machine Tools
Proc. IEE, Vol 116, No. 11. 1969. nov.
- [60] HENDERSON, W.T.K. The 2CL Programme for Numerical
Control
Machinery and Production Engineering 1967. okt. 18.
- [61] HERMANSON, A.E. Computer Machining on Line
American Machinist, 1969. aug. 25.
- [62] HIRSCH, C.O. CAD/CAM and Government
Society of Manufacturing Engineers, Technical
Papers No. MS 72-905 1972.

- [63] HORVÁTH, M.-NAGY, S. NC-esztergák számítógépes programozási rendszereivel szemben támasztott követelmények
Compcontrol'72 Konferencia Sopron 1972. jun. 19-24.
- [64] IREDALE, R. Today's Machining Centres Make Tomorrow's Manufacturing Systems
Metalworking Production, 1969. dec. 3.
- [65] JOHNSON, T.E. Sketchpad III. - A Computer Program for Drawing in Three Dimensions
Proc. Spring Joint Comp. Conf. 1963.
- [66] KARDOS, P. Programvezérlésű Numerikus Pályagenerátor
MTA SZTAKI Tanulmány 1967.
- [67] KARPLUS, W.J. Time-Shared Man-Computer Systems
On-Line Computing McGraw-Hill, Inc. New York 1967.
- [68] KELLEY, R.A. APT-ADAPT
American Machinist 1964. jun. 22.
- [69] KENNEDY, I.R. A System for Time-Sharing Graphic Consoles
Proceeding-Full Joint Computer Conf. 1966.
- [70] KLAUS, R.L. New Concepts in Interactive Computing and Their Relationship to Computer-Aided Design
IFIP Working Conference on Principles of CAD
Eindhoven, 1972. okt. 16-18.
- [71] KIRSCH, R.A. Computer Interpretation of English Text and Picture Patterns
IEEE Transactions on Electronic Computers
1964. aug.
- [72] KOLOC, J. The Use of Workpiece Statistics to Develop Automatic Programming for NC Machine Tools
Int. I.Mach Tool Des.Res Vol 9. p. 65-80 1969.

- [73] KOLOR, J. New Method of Programming NC Machine Tools Machinery Lloyd (European Edition) 1968. dec.
- [74] KOVÁCS, M. EXAPT1 - A pont és szakaszvezérlésű szer-
számgepek szimbolikus programnyelve
MTA SzTAKI Közlemények 2. sz. 1969.
- [75] KRAMMER, G. EXAPT1 rendszer honosítási problémái
Compcontrol'70 Konferencia Miskolc 1970.
- [76] KRITZAN, I. CRT Programming Cuts Cost.
American Machinist 1970. szept. 21.
- [77] LACOSTE, J.P. Communication in Computer Aided Design
and Computer Aided Manufacturing
Prolamat'73 Conference Budapest, 1973. ápr.10-13.
- [78] LAPIDUS, G. A Look at Minicomputer Applications
Control Engineering 1969. nov.
- [79] LESLIE W.H.P. Features of Computer Programs for
Numerical Control
Proceeding-Full Joint Computer Conf. 1969.
- [80] LICKLIDER, J.C.R. Man-Computer Simbiosis
IRE Trans. Human Factors in Electronics HFE-1
1960. márc.
- [81] LICKLIDER, J.C.R. On-Line Man-Computer Communication
Proc. of the SJCC. San Francisco 21. 1962. máj.
- [82] LICKLIDER, J.C.R. Principles and Problems of Console
Design
Proc. of the IFIF Cong. 65.
- [83] LOW, A.H. An Introduction to Metal-Cutting Technology
in NC Languages
Prolamat Conference Roma 1969. szept. 15-17.
- [84] MANGOLD, W.E. Status of NC Language Standardization in
I.S.O.
Prolamat Conference Roma 1969. szept. 15-17.

- [85] MANGOLD, W.E. APT Part Programming of the Sculptured Surface
American Society of Tool and Manufacturing Engineers, Technical Paper MS68-117 1968.
- [86] MARKLEW, J.J. Developments in the System Operated by Olivetti in Their Italian Toolrooms
Machinery and Production Engineering, 1970. jan. 28.
- [87] MARKLEW, J.J. Advanced ideas for Production in the 70's at Ferranti, Edinburgh
Machinery and Production Engineering 1970. febr. 4.
- [88] MARKLEW, J.J. First Kearns-Richards 45 PT Machining centre in Operation on DS & G Lathe Headstock Castings.
Machinery and Production Engineering, 1969. szept. 10.
- [89] MÁRKUS, T. Tervezőprogram vezértárcsák NC megmunkáláshoz
Compcontrol'72 Konferencia Sopron 1972.
- [90] MILLER, B.M. Look to Time-Sharing in Preparing NC Tapes Automation, 1969. aug.
- [91] MITTMAN, B. Development of Numerical Control Programming Languages in Europe
Proceedings A.C.M. National Meeting 1967.
- [92] MOTARD, R.L. Program and Information Structures in Computer-Aided Design
IFIP Working Conference on Principles of CAD Eindhoven 1972. okt. 16-18.
- [93] NEMES, L. Szerszámgépek számítógépes irányítása
MTA SzTAKI Tanulmányok 1972.

- [94] OPITZ, H.-BERGER, H. Calculation of Technological Data for the Computer-Assisted Programming of Numerically Controlled Machine Tools
Prolamat Conference Roma, 1969. szept. 15-17.
- [95] OPITZ, H.-SIMON, W. The Programming of Numerically Controlled Machines with EXAPT
Machinery and Production Engineering 1967. aug.
- [96] PECK, E.S. NC Today: Directions in Advanced Manufacturing Techniques
Manufacturing Engineering & Management 1970. nov.
- [97] PECK, E.S. NC at the Eleventh European Machine Tool Exposition
The Tool and Manufacturing Engineer 1969. okt.
- [98] PETERSON, C. For NC - A Management Philosophy That Works.
Manufacturing Engineering & Management 1970. jan.
- [99] PIKLER, GY. EXAPT1 Számítógépes programozási nyelv és honosítási feladatai, tapasztalatai
Compcontrol Konferencia Miskolc 1968.
- [100] PIKLER, GY. 2CL szimbolikus nyelv NC marógépek programozásához
MTA SzTAKI Közlemények 2. sz. 1969.
- [101] PIKLER, GY. MUS-320 furómarómű EXAPT1 post-processzora
Compcontrol'70 Konferencia Miskolc 1970.
- [102] PIKLER, GY. Számjegyvezérlésű szerszámgépek automatikus (számítógépes) programozása
Gépipari Tudományos Egyesület 84/1971. Kiadványa 1971.
- [103] PIKLER, GY. A Minicomputer-Based, Conversational Program Writing System
Prolamat'73 Conference Budapest 1973. ápr.10-13.

- [104] PRUUDEN, J. APROKS. A Programming System for NC Flame Cutters
Prolamat Conference Roma 1969. szept. 15-17.
- [105] RALPH, G. The Critical Years
The Production Engineer, 1968. jun.
- [106] REASON, J. Computer Outdate Hard-wired Control Experts Speak Out
Control Engineering 15 No. 1. 1968. jan.
- [107] RECKZIEGEL, D.-GRUPE, U. The Programming System of EXAPT
Prolamat Conference Roma 1969. szept. 15-17.
- [108] RIZOR, M.L. A Vote for In-House Programming
Manufacturing Engineering & Management 1970. nov.
- [109] ROBINSON, F. Future Trends in Numerical Control
The Production Engineer 1968. szept.
- [110] SABIN, M.A. Computer-Aided Part Program Generation
Prolamat'73 Conference Budapest 1973. ápr.10-13.
- [111] SCREASE, T. Controlling Numerical Control
Metal Working Production 1970. márc. 4.
- [112] SIDERS, R.A. Computer Graphics. A Revolution in Design
American Management Association INC. New York
1966. (pp. 160)
- [113] SIDERS, R.A. Computer Aided Design
IEEE Spectrum 1967. nov.
- [114] SIDERS, P.A. Machining Precision Components Under
Program and Tape Control
Machinery and Production Engineering 1970.márc.4.
- [115] SIM, R.M. Numerically-Controlled Machine Tool Programming
Machinery and Production Engineering 1965.dec.22.
- [116] SIMON, R. Rechnerunterstütztes Konstruieren
Industrie-Anzeiger 90. 1968. okt.

- [117] SIMON, W. Perspektiven des Einsatzes von D.V.A. in der europäischen Fertigungstechnik
Elektronische Rechenanlagen 9. 1967.
- [118] SCHLENIUS, G. Workshop Technology, Especially Turning
Prolamat Conference Roma 1969. szept. 15-17.
- [119] SPUR, G.-TANNENBERG, F. Die Programmierung Numerisch Gesteuerter Drehmaschinen mit EXAPT2
Metallbearb. 62. 1968.
- [120] STUTE, G. The Milling Technology in EXAPT3.
Prolamat Conference Roma 1969. szept. 15-17.
- [121] SUTHERLAND I.E. SKETCHPAD - A Man-Machine Graphical Communication System
Proc. Spring Joint Comp. Conf. 1963.
- [122] TAMM, B.G. On the Data Structure and Linkage Problems in NC Machine Tool Languages
Prolamat Conference Roma 1969. szept. 15-17.
- [123] TANKO, J. Tanulmány a numerikus vezérlésű szerszámgépek automatikus programozásáról
MTA Számítástechnikai Központja 1966.
- [124] TAYLOR, R.M. Man-Computer Input-Output techniques
IEEE Trans. on Human Factor in Electronics
1967. márc.
- [125] THORNHILL, D.E. A Sample Interactive Graphics Program
NEL Course Notes 1968. aug.
- [126] THORNHILL, R.B. Numerical Control Terminology
Automation 1968. szept.
- [127] TIMMS, C. Machine Tool Development Sponsored by the Ministry of Technology
The Production Engineer 1968. jun.
- [128] TIPTON, H. Information and the Control of Machine Tools
The Production Engineer 1968. szept.

- [129] WEILL, R. IFAPT: A Unified System of Modular Design
for NC Languages
Prolamat Conference Roma 1969. szept. 15-17.
- [130] WICK, C.H. Economical Direct Computer Control for
Machine Tools
Machinery and Production Engineering
1970. jun. 24.
- [131] WIENDAHL, H.P. Computer Aided Design in Machine
Building Industry
International Conference for CAD Southampton
1972. ápr. 24-28.
- [132] WILKINSON, D.G. Conversational NALAPT, an Interactive
Version of the NELAPT Processor
Prolamat'73 Conference Budapest 1973. ápr.10-13.
- [133] WILKINSON, D.G. Számítógépes tervezés és gyártás a
gépiparban
CAD Centre-MTA SzTAKI Szeminárium Budapest
1973. április 16-18.
- [134] UZSOKI, M.-PASZTOR, K. Computer Aided Design of Digital
Systems in Hungary
International Conference for CAD Southampton
1972. ápr. 24-28.
- [135] ZEHNPFENNIG, W.C. Automatic Positioning by NC
Machine Design 1970. máj. 14.
- [136] YOUNG, N.F. Distributed Computer Systems
Automation 1969. okt.
- [137] Computer System ADAPT
Control Data Corporation No. 6017340 1966. szept.
- [138] APT Part Programming
The APT Long Range Program Staff, IIT Research
Institute 1967 (MCGrow-Hill Book Comp.)

- [139] The Chicago Machine Tool Show-1970
Machinery and Production Engineering,
1970. okt. 21.
- [140] 2CL Part programming Reference Manual
NEL Report No. 299 Minister of Technology 1967.
- [141] Systeme de programmation IFIPT P point a point
Compagnie Internationale Pour L' Informatique
- [142] Systeme de programmation IFAPT C Continu
Compagnie Internationale Pour L' Informatique
- [143] Systeme de programation IFAPT
Compagnie Internationale Pour L' Informatique
- [144] Computer-Aided Design
NEL Report No. 242 1966.
- [145] AT Last - A Computer with heart (New Products)
Electronics, 1969. jan. 6.
- [146] Digital Computer Graphics
American Data Processing. Inc. Detroit 1967.
- [147] 11 th European Machine Tool Exhibition, Paris-8
Machinery and Production Engineering
1969. szept. 24.
- [148] 11 th European Machine Tool Exhibition Paris-10
Machinery and Production Engineering
1969. okt. 27.
- [149] 11 th European Machine Tool Exhibition Paris-11
Machinery and Production Engineering
1969. nov. 19.
- [150] EXAPT1 Part Programmer Reference Manual
NEL Report No. 293 1967. jul.
- [151] EXAPT2 Part Programmer Reference Manual
EXAPT-Verein Aachen, Best-Nr. 69. 06. 01.

- [152] EXAPT2 Processor Description
EXAPT-Verein Aachen. Order-No 69. 04. 05.
- [153] Introduction to Data Processing for Numerical Control
of Machine Tools
IBM Data Processing Application 1967.
- [154] Low Cost Computer-Aided NC Tapemaking
Machinery 1969. okt.
- [155] Low-Cost Man-Computer Interfaces? Not yet!
Electro-Technology, 1969. febr.
- [156] More than 2 Million Characters
American Machinist 1969. jun. 16.
- [157] Conversational 2CL
NEL Közlemények NC news No. 4. 1970.
- [158] Programming of Numerically Controlled Machine Tools
NEL Report No. 187
- [159] New Programming Aid Sumplifies NC Machining
Machinery 1969. szept.
- [160] No Cards Needed to Prepare NC Tapes
Control Engineering 1970. jan.
- [161] Numerikus vezérlésű szerszámgépek program leírása
MTA SzTAKI Tanulmány 1972.
- [162] Conversational MINIAPT System, Keyboard Conversation/
Graphic Conversation
OKI Electric Ind. Co. Ltd.
- [163] Számjegyes szerszámgépvezérlés legújabb műszaki fej-
lesztési irányai
OMFB Konceptiótervezet 14-402/d-Kt 1969. jun.
- [164] Point-to-Point NC System
Metalworking Production, 1970. márc. 25.

- [165] Rolls-Royce Order Molins System 24
Metalworking Production 1969. febr. 2.
- [166] Some New Russian Machine Tools
Machinery and Production Engineering,
1970. jun. 17.
- [167] Staveley Introduces Moduleplan range of machining
centres
Machinery and Production Engineering
1969. dec. 10.
- [168] System/360 AUTOSPOT Numerical Control Processor
IBM Application Program No H20-0179-1
- [169] System/360 ADAPT/AUTOSPOT Numerical Control Processor
IBM Technical Newsletter No H20-1065-0
- [170] US Adds Technology to Turning Programmes
Metalworking Production 1969. márc. 12.
- [171] A számítógépes műszaki tervezés
OMFB Tanulmány 16-803/6-Et 1972. febr.
- [172] Burkner-Ramo type 2601 and 3601 NC Systems for Lathes
Machinery and Production Engineering
1969. dec. 24.

AJÁNLOTT IRODALOM

- [173] ANDERSSON, A.K. Macrotechnik i APT och ADAPT
Verkstäderna 1970. nov.
- [174] BEAR, J.L. Frequency Numbers Associated with Directed
Graph Representations of Computer Programs
Proc. 2 nd Hawaii Internat. Conf. on Syst.
Sci. 1969.

- [175] COCKE, J. Some Analysis Techniques for Optimizing
Computer Programs
Proc. 2 nd Hawaii Internat. Conf. on Syst.
Sci. 1969.
- [176] ENCARNACAO, J. Das interaktive Entwerfen und Zeichnen
in drei Dimensionen auf Computer-Bildschirmgeräten
Elektronische Rechenanlagen Heft 1 1971. nov.
- [177] FERENCZY, J. Sajtolószerszámok és sajtolási technológia
számítógépes interaktív tervezése
MTA SzTAKI Közlemények 8. sz. 1969.
- [178] HATVANY, J. Computer Aided Design
CAD Centre-MTA SzTAKI Szeminárium Budapest
1973. április 16-18.
- [179] HILL, D.R. Some Practical Steps Taken Towards a Man-
Machine Interface Using Speech
Proc. 2 nd. Hawaii Internat. Conf. on Syst.
Sci. 1969.
- [180] KEEFE, O.K. Modular Sequential Machines
Proc. 2 nd. Hawaii Internat. Conf. on Syst.
Sci. 1969.
- [181] KLERER, M. Computer Language Structure and Time-Sharing
Operation
Proc. 2 nd. Hawaii Internat. Conf. on Syst.
Sci. 1969.
- [182] KHOL, R. Adaptive Control Toward the Thinking Machine
Machine Design 1969. máj. 1.
- [183] KRAMMER, G. FLISP 0.5. Egy listakezelő fortran szubru-
tin rendszer
MTA SzTAKI Tanulmány 1971. nov.
- [184] LESLIE, W.H.P. NC alkalmazási Kézikönyv
Műszaki Könyvkiadó 1973.

- [185] PIKLER, GY. Interaktiv sajtólószerszám tervező rendszer
CAD Centre-MTA SzTAKI Szeminárium Budapest
1973. ápr. 16-18.
- [186] PIKLER, Gy. Hidegsajtoló szerszámok számítógépes in-
teraktiv tervezése
Magyar Híradástechnikai Egyesülés Tanulmány 1973.
- [187] SILACCI, R.C. Computer Graphics Application to Computer
Aided Logic Design
Proc. 2 nd. Hawaii Internat. Conf. on Syst.
Sci. 1969.
- [188] SMITH, L.B. Interactive Graphical Systems for
Mathematics, A Survey
Proc. 2 nd. Hawaii Internat. Conf. on Syst.
Sci. 1969.
- [189] ZINN, K.L. Instructional Uses of Interactive Computer
Systems
Datamation 1968. szept.
- [190] YAMAGUCHI, F. Graphic APT System-Dialogue-Type
Technical Research Institute Report. Japan 1971.
- [191] Preliminary Development of Purdue's Remote Interactive
Design System
Purdue University Report 1970. jun. 30.
- [192] HERZAG, B. Draw L 70: Computer Graphics Language
The University of Michigan Technical Report 30
1970. aug.
- [193] SYMAP (B) Symbolprache zur maschinellen Programmierung
von Bahnsteuerungen
VEB Carl Zeiss JENA 1967.

- [194] MARTIN, J. Design of Man-Computer Dialogue
Prentice-Hall Inc. Englewood Cliffs, N.J. 1973.
- [195] Conversational NELAPT
NEL NC Program Notes No 4 1973.
- [196] A mérnöki munka szervezéstechnikai eszközei
OMFB Tanulmány (15-7301 IT)

A TANULMÁNYOK sorozatban eddig megjelentek:

- 1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundáns, $\{\wedge, \vee, \neg\}$ vagy $\{\text{NOR}\}$ vagy $\{\text{NAND}\}$ bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítása
- 2/1973 Вашкеви Иштван: Расчленение многосвязных промышленных процессов с помощью вычислительной машины
- 3/1973 Ádám György: A számítógépipar helyzete 1972 második felében
- 4/1973 Bányász Csilla: Identification in the presence of drift
- 5/1973^{*} Gyürki J.-Laufer J.-Girnt M.-Somló J.: Optimalizáló adaptív szerszámgépirányítási rendszerek
- 6/1973 Szelke Erzsébet-Tóth Károly: Felhasználói Kézikönyv /USER MANUAL/ a Folytonos Rendszerek Szimulációjára készült ANDISIM programnyelvhez
- 7/1973 Legendi Tamás: A CHANGE nyelv/multiprocesszor
- 8/1973 Klafszky Emil: Geometriai programozás és néhány alkalmazása
- 9/1973 R.Narasimhan: Picture Processing Using Pax
- 10/1973 Dibuz Ágoston-Gáspár János-Várszegi Sándor: MANU-WRAP hátlaphuzalozó. MSI-TESTER integrált áramköröket mérő, TESTOMAT-C logikai hálózatokat vizsgáló berendezések ismertetése
- 11/1973 Matolcsi Tamás: Az optimum-számítás egy új módszeréről

- 12/1973 Makroprocesszorok, programozási nyelvek. Cikkkgyűjtemény az NJSzT és SzTAKI közös kiadásában. Szerkesztette: Legendi Tamás
- 13/1973 Jedlovsky Pál: Uj módszer bonyolult rektifikáló oszlopok vegyészmérnöki számítására
- 14/1973 Bakó András: MTA Kutatóintézeteinek bérszámfejtése számítógéppel
- 15/1973 Ádám György: Kelet-nyugati kapcsolatok a számítógépiparban
- 16/1974 Fidrich Ilona-Uzsoky Miklós: LIDI-72 Listakezelő rendszer a Digitális Osztályon 1972. évi változat
- 17/1974 Gyürki József: Adaptiv termelésprogramozó rendszer /APS/ termelő műhelyek irányítására

*-gal jelölt kivételével a TANULMÁNYOK megrendelhetők az Intézet Könyvtáránál /Budapest, I. Uri u. 49./

Felelős kiadó:

MTA Számítástechnikai és Automatizálási
Kutató Intézet

Jelen dolgozat a 3.6.3 "Számítógépek alkalmazása szer-
számgepek vezérléséhez" c.
intézeti alapkutatási téma
keretében készült.

Beérkezett: 1974. I. 7.

